

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Modélisation en Intelligence Artificielle

Zampelli, Stéphane

Award date:
2002

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année Académique 2001 - 2002
Promoteur : Professeur Jacques Berleur s.j.

Modélisation en Intelligence Artificielle
Zampelli Stéphane

Mémoire présenté en vue de l'obtention du grade de Maître en
Informatique

UBS 10102541

Abstract

This thesis took place in the aera of multiagents systems in the field of artificial intelligence (AI) ; in particular we worked on a multiagents system called Agency, developped at the Politecnico di Milano.

The technical part is centered on the comprehension of this machine, which is able to incorporate different models from a same phenomenon in order to represent this phenomenon. In particular, we have worked around an *anthropic* agency which aims to replace any organ in the human body by using partial models. Our study case was the replacement of the physiological process of the pancreas, in order to solve the diabetic pathology.

The epistemological part of this thesis, which is the main part, tries to understand the epistemological difference between empirical sciences and AI. We have been helped in our research by the example of the agency, which is supported by a very developped epistemological framework. Our thesis is that AI eliminates the transcendance of the objet and changes the link between subject and object.

Résumé

Cette thèse se situe dans le domaine des systèmes multi-agents dans le champ de l'intelligence artificielle (IA) ; en particulier nous avons travaillé sur un système multi-agents appelé Agence, développé au Politecnico di Milano.

La partie technique est centrée sur la compréhension de cette machine, qui est capable d'incorporer différents modèles d'un même phénomène en vue de représenter ce phénomène. En particulier, nous avons travaillé sur une agence anthropique qui vise à remplacer n'importe quel organe dans le corps humain en utilisant des modèles partiels. Notre cas d'étude était le remplacement du processus physiologique du pancréas pour résoudre la pathologie du diabète.

La partie épistémologique, qui est la partie principale, essaye de comprendre la différence épistémologique entre les sciences empiriques et l'IA. Nous avons été aidés dans notre recherche par l'exemple de l'agence, qui est soutenue par un cadre épistémologique très développé. Notre thèse est que l'IA élimine la transcendance de l'objet et change radicalement le lien entre sujet et objet.

Mes remerciements pour m'avoir permis de réaliser ce travail vont
à ma mère, mon père, toute la famille Rancati, ma grand-mère
Bianca, Rossi Adriana, Zampelli Italia, Delval Quentin, Masure Sylvie.

Mes remerciements vont également au Professeur Jacques Berleur
s.j. pour son soutien durant l'élaboration de ce mémoire, et également au
Professeur Somalvico et son équipe pour leur accueil à Milan.

Je tiens également à remercier le Père Ganty de la faculté de Philosophie
de Namur pour son soutien durant mes études en philosophie.

Quant à ceux qui se sentent oubliés, qu'ils sachent que je pense à eux.

Table des matières

Introduction	6
I Agence anthropique	10
1 Le concept d'Agence	11
1.1 Introduction	11
1.2 Les Agents	11
1.3 Les systèmes multi-agents	12
1.4 La notion d'Agence	14
1.4.1 L'Agence au sein des systèmes multi-agents	14
1.4.2 L'Agence comme une machine de coopération	17
1.5 Classification des Agences	18
1.6 L'aporie de la tout de Babel	19
1.7 Conclusion	20
2 L'agence anthropique	22
2.1 Introduction	22
2.2 Architecture générale	22
2.2.1 Développement de l'architecture	23
2.2.2 Description de l'architecture	30
2.3 Les agents, les tableaux et le majordome	32
2.3.1 Agent extracteur	32
2.3.2 Agent décisionnel	35
2.3.3 Agent actuateur	40
2.3.4 Parameter BlackBoard	44
2.3.5 Knowledge BlackBoard	45
2.3.6 Majordome	46
2.4 Le mécanisme de coordination	49
2.4.1 Spécification du problème	49
2.4.2 Idée de la solution	50
2.4.3 Objectif commun	52

2.4.4	Attribution des actions	55
3	Générateur de représentation de modèles physiologiques pour une approche distribuée	58
3.1	Introduction	58
3.2	Spécification du problème	58
3.3	Modèle physiologique	59
3.4	Génération de la représentation du modèle	61
3.4.1	Espace multi-dimensionnel	61
3.4.2	Génération des solutions	62
3.4.3	Degré de pathologie	62
3.4.4	Introduction des données	63
3.5	Exemples de modèles	65
3.6	Conclusion	65
II	Démarche épistémologique	67
4	Découverte scientifique	68
4.1	Introduction	68
4.2	La découverte scientifique	68
4.2.1	Le débat	68
4.2.2	Cadre épistémologique de la découverte scientifique . .	69
4.2.3	Double rôle de la machine dans la découverte scientifique et leur interaction	74
4.3	Architecture pour la description de la découverte scientifique .	77
4.3.1	La métaphore de la mosaïque	77
4.3.2	Agence représentationnelle	78
4.3.3	Différents niveaux d'abstraction	80
4.3.4	Innovation dans la découverte scientifique	81
4.4	Conclusion	85
5	Le concept de représentation	87
5.1	Introduction	87
5.2	La représentation	87
5.3	Le rôle herméneutique de la notion de représentation	88
5.3.1	La genèse de la théorie de la représentation	89
5.3.2	Les origines cartésiennes de la notion de représentation	92
5.3.3	Conclusion : la représentation dans le paradigme de l'ordinateur	93
5.4	Conclusion	93

6	Le concept de modèle	95
6.1	Introduction	95
6.2	La fonction du modèle	95
6.3	Le modèle dans les sciences empiriques	97
6.4	Sciences empiriques et ordinateur	98
6.4.1	Spécificité de la modélisation biologique	99
6.4.2	L'expérience de pensée	103
6.4.3	Nature de l'ordinateur	105
6.4.4	Extériorité interne de la pensée	106
6.4.5	Intelligence Artificielle et sciences empiriques	107
6.4.6	Le double rôle de la machine universelle de Turing . . .	109
6.5	Application de ces concepts	109
6.5.1	Le cas concret de l'agence anthropique	109
6.5.2	La contrainte du discret dans la représentation	111
6.5.3	Le double rôle de l'Agence Anthropique	113
6.5.4	Signification épistémologique de l'agence anthropique .	116
	 Conclusion	 117

Introduction

Les systèmes multi-agents deviennent aujourd'hui en intelligence artificielle une tendance marquée. Nous avons eu la chance de pouvoir travailler avec une équipe de chercheurs à Milan qui tentent de créer un nouveau type de système multi-agents, l'agence. Cette idée d'agence est basée sur le livre de Minsky «La société de l'esprit», qui affirme que la connexion d'éléments non-intelligents est la clé pour donner lieu à des entités plus grandes qui possèdent elles des capacités cognitives. En particulier, nous avons travaillé sur une agence appelée *agence anthropique*. Sa particularité, comme son nom l'indique, est de s'appliquer au corps de l'homme. L'objectif à long terme de cette agence est de *remplacer* le pancréas. Cette agence anthropique servirait en effet de système de contrôle pour l'insuline, en s'attaquant dans un premier temps au cas du diabète. Sa particularité, en tant que système multi-agents, est d'être capable d'intégrer différents modèles a priori non reliés entre eux et de les faire coopérer pour réguler le corps. Concrètement, cela signifie que l'on irait chercher des modèles métaboliques recouvrant le problème du diabète, qu'on les transformerait dans des modèles manipulables par les agents de l'agence anthropique, et qu'on les intégrerait dynamiquement dans l'agence. Ces agents, en première approximation, peuvent être représentés comme des unités de calcul capables de gérer un modèle et de se coordonner avec les autres agents en vue d'atteindre un objectif commun.

C'est dans ce cadre technique que nous avons alors entamé une réflexion épistémologique sur la nouveauté qu'introduit l'intelligence artificielle dans le rapport sujet/objet. La question de départ était la suivante : quelle est la spécificité méthodologique de l'intelligence artificielle par rapport aux sciences empiriques classiques ? Notre préoccupation était d'abord épistémologique avant d'être technique, mais cela nous a donné l'occasion de confronter les deux niveaux. L'intérêt de cette question épistémologique est par ailleurs soulignée par le but même de l'agence anthropique. Car à l'heure où bon nombre de scientifiques remplacent des parties du corps par un processeur, il devient urgent de se demander quel rapport au monde ces pratiques engendrent. On voit par exemple certaines parties du cerveau reliées en entrée à une capacité de traitement d'information ; celle-ci envoie des influx nerveux en sortie dans le cerveau, et l'on essaye le plus souvent d'obtenir des réactions cohérentes. Une telle fusion entre machine à informations et corps semble bien sûr promettre de nouvelles voies à la médecine, mais il nous faut comprendre ce qui est exactement en train de se produire. Nous proposons une réponse à ces questions, en montrant comment l'intelligence artificielle modifie le rapport entre sujet et objet.

Le plan de ce travail se décompose donc en deux : d'une part une partie technique, d'autre part une partie épistémologique.

La partie technique est structurée comme suit.

Le chapitre 1 présente les systèmes multi-agents et le concept d'agence. Nous essayons d'y définir la notion d'agent, compris comme une unité non élémentaire offrant certaines fonctionnalités. Nous insisterons sur le flou flottant autour de cette notion d'agent, principalement sur la question de savoir ce qu'est une unité «non élémentaire». Nous définissons alors le concept de système multi-agents, qui met l'accent sur l'*interconnexion* entre les agents. Nous progresserons ensuite vers une définition du concept d'agence. L'agence est un système multi-agents particulier, qui s'en distingue par des caractéristiques précises de ses agents. Son originalité est de faire coopérer des modèles d'un même phénomène, les agents possédant chacun un modèle partiel du phénomène. L'agence apporte également une innovation dans la conception des systèmes multi-agents en minimisant le problème d'intégration dynamique des agents, problème mieux connu dans la littérature sous le nom de l'«aporie de la tour de Babel». Le concept général d'agence est ainsi introduit.

Nous passons ensuite dans le chapitre 2 à la description technique de l'agence sur laquelle nous avons travaillé, l'agence anthropique, où nous montrons son architecture interne et nous décrivons complètement son fonctionnement. L'agence anthropique est le concept d'agence appliqué au corps. L'enjeu de cette agence anthropique est de construire une architecture *générique*, capable d'être appliquée à diverses parties du corps. Cette agence anthropique devra être complètement dynamique, c'est-à-dire que les agents pourront être intégrés durant le fonctionnement de l'agence anthropique. Trois types d'agents sont présents : des agents extracteurs qui prélèvent les signaux du corps, des agents décisionnels qui évaluent l'état à partir de leur propre modèle, et enfin les agents actuateurs qui réalisent les actions adéquates dans le corps. L'originalité de cette agence anthropique provient de ce que chaque agent possède un modèle partiel de l'organe remplacé ; l'agence anthropique est alors capable de les faire coopérer pour obtenir une décision unique et cohérente. De plus, un modèle partiel (incorporé à un agent décisionnel) peut être intégré dynamiquement au sein de l'agence anthropique de telle sorte que l'on peut imaginer des bibliothèques de modèles que l'on insérerait et supprimerait dynamiquement. La partie la plus critique dans cette agence anthropique est le mécanisme de coordination, qui doit faire se coordonner les différents agents décisionnels pour obtenir une décision commune prenant en compte les exigences de chaque

modèle partiel. Le développement de cette architecture nous donne ainsi un exemple concret d'une implémentation d'une agence.

Dans le chapitre 3 nous décrivons un générateur de modèle, qui convertit les modèles mathématiques en modèles utilisables par l'agence anthropique. Les modèles qui seront initialement utilisés dans l'agence anthropique sont des modèles physiologiques mathématiques. Or ces modèles physiologiques possèdent des particularités qu'il faut concilier avec une représentation au sein de la machine. Principalement, les modèles physiologiques sont dynamiques, c'est-à-dire que la sortie de ces modèles ne dépend pas seulement de l'entrée mais également de paramètres externes. Il a fallu faire ainsi une passerelle logicielle entre les modèles mathématiques, physiologiques, et dynamiques d'une part et la représentation interne à un agent décisionnel d'autre part, statique et devant se conformer aux critères de recherche des agents décisionnels.

La partie épistémologique commence avec le chapitre 4 par un exposé de la position épistémologique des chercheurs de Milan, qui sont engagés, sur le plan épistémologique, dans le débat de la découverte scientifique. Rappelons que ce débat pose la question suivante : quel est le rôle des machines dans l'activité scientifique ? Sont-elles capables de mener cette activité scientifique ou ne sont-elles d'aucune utilité dans la recherche scientifique ? L'agence, outre son intérêt purement technique, a également une visée épistémologique, puisque l'équipe de Milan semble vouloir lui faire jouer un nouveau rôle dans la découverte scientifique. Outre un rôle de soutien plus classique, l'agence pourrait prendre un rôle de description du résultat de la découverte scientifique. L'équipe de Milan propose à cet égard une agence particulière, appelée agence représentationnelle, dans laquelle seraient représentés tous les actes cognitifs de la découverte scientifique, comme l'innovation, la généralisation, la simplification, etc. ainsi que les liens entre les différents modèles.

Avec la chapitre 5 nous commençons notre réflexion sur la modélisation en intelligence artificielle en insistant sur le terme de représentation. Nous y verrons qu'en intelligence artificielle la représentation se fait dans le cadre du discret, et qu'elle se différencie par là de la représentation en sciences empiriques. Ces réflexions générales sur le cadre épistémologique de l'intelligence artificielle sont importantes pour comprendre sous quels pré-supposés l'intelligence artificielle aborde l'étude de son objet. Ce cadre n'est pas neutre, il influence par exemple la manière dont on aborde la découverte scientifique et la représentation des modèles au sein de l'agence

comme nous le montrerons dans le chapitre 6.

Le chapitre 6 quant à lui explore la notion de modèle et l'acte de modélisation en intelligence artificielle. En particulier, nous essayons de déterminer en quoi consiste au juste la modélisation dans les sciences empiriques. A partir de là nous essayons alors de différencier l'intelligence artificielle de cette méthode de modélisation en science, et nous répondons alors à notre question de départ tout en jetant un regard critique sur les thèses épistémologiques de Milan. Nous défendrons la thèse selon laquelle ce qui différencie intelligence artificielle et sciences empiriques est la disparition de la transcendance de l'objet. Alors que les sciences empiriques font face à une certaine réalité, qui sert de contrôle par rapport à la théorie, l'objet de l'intelligence artificielle est l'ordinateur faisant fonctionner un programme, c'est-à-dire une partie de la pensée de l'homme. Il n'y a plus alors de transcendance de l'objet, puisque l'objet est la re-production de la pensée du sujet. Sur base cette différenciation entre intelligence artificielle et sciences empiriques, nous adresserons alors quelques critiques sur le point de vue épistémologique de Milan, notamment en ce qui concerne le rôle de description de l'agence.

Première partie

Agence anthropique

Chapitre 1

Le concept d'Agence

1.1 Introduction

L'objectif de cette partie est de situer et de définir le concept d'agence dans le domaine plus général de l'intelligence artificielle appelé Distributed Artificial Intelligence.

1.2 Les Agents

Le domaine de l'intelligence artificielle appelé Distributed Artificial Intelligence se propose d'étudier les systèmes dans lesquels diverses entités, appelées agents, sont distribuées dans l'espace mais connectées entre elles. La granularité des entités distribuées permet de distinguer deux types de systèmes, même si les critères de granularité sont arbitraires. Les instructions d'un microprocesseur, par exemple, pourraient être considérées comme des entités fines. Lorsque la granularité est fine, on parle de systèmes parallèles. Lorsque l'on obtient une granularité suffisante, les entités sont alors appelées agents et les systèmes correspondants des systèmes multi-agents. Malheureusement, il ne semble pas y avoir de définition précise du concept d'agent. Un agent, d'une manière générale, peut être tout système interactif, comme un processus logiciel, un ordinateur ou un robot. F. Amigoni [4, p. 9] propose, pour clarifier la situation, une classification de la notion d'agent selon quatre axes :

- Paradigme : un agent est défini sur la base de son architecture conceptuelle. On peut ainsi définir un agent comme une unité d'exécution, composé du code, d'un espace de données et d'un espace d'exécution. Un processus classique peut donc être compris comme un agent selon

cette première dimension.

- Formalisme : un agent est défini selon sa description formelle. Par exemple un agent peut être défini dans cette dimension comme un système inférentiel.
- Environnement : un agent est défini sur la base des outils logiciels ou matériels utilisés pour son développement ou pour son exploitation. Par exemple un agent peut être défini dans cette dimension comme un objet Java.
- Application : un agent est défini sur la base du domaine auquel il s'applique. Par exemple, un agent peut être défini comme un assistant personnel qui collabore avec l'utilisateur dans un environnement de travail déterminé.

Cette classification des agents met en avant l'hétérogénéité des définitions des agents. Un agent défini selon une dimension peut apparaître selon une autre dimension comme n'étant pas un agent. Dès lors que l'on entend exploiter les systèmes multi-agents, il devient important de situer exactement à quelle dimension appartiennent les agents du système. On peut en effet obtenir une classification des systèmes multi-agents à partir de la nature des agents, comme nous le verrons dans la section suivante.

1.3 Les systèmes multi-agents

Nous voudrions dans cette section présenter le concept de système multi-agents, dont l'agence fait elle-même partie.

Remarquons tout d'abord l'importance de la connexion entre les entités appelées agents. Puisque la granularité peut être difficilement définie, et puisque la définition même de la notion d'agent semble problématique, le seul élément consistant de la définition des systèmes multi-agents est l'interconnexion entre ceux-ci. On voit d'emblée l'enjeu qui se profile derrière cette définition des systèmes multi-agents : l'objet principal des systèmes multi-agents est l'interconnexion entre des entités distinctes. Alors que le paradigme classique de l'informatique consiste à prendre en compte une entité principalement composée d'entrées et de sorties, il s'agit avec les systèmes multi-agents de considérer l'interconnexion de diverses entités.

On parle de système multi-agents lorsqu'il s'agit d'un système du domaine de l'intelligence artificielle distribuée et lorsque la granularité est suffisante. Un système multi-agents se définit donc comme un système consistant en une interconnexion d'agents.

Nous exposons à présent dans cette section une taxonomie des systèmes

multi-agents, proposée par Amigoni [4]. On pourrait se demander à juste titre l'utilité d'une telle taxonomie alors que notre objectif est de définir le concept d'agence. En fait l'agence est un système multi-agents, et elle n'échappe pas à la taxonomie qui suit et que nous utiliserons également dans le cas de l'agence. Des quatre dimensions permettant de définir la notion d'agent, on peut déduire quatre dimensions similaires permettant de caractériser un système multi-agents :

- Paradigme : le système multi-agent est caractérisé par les architectures conceptuelles et les structures des agents ainsi que par leurs interactions, par exemple un système multi-agents où les agents sont décrits comme une hiérarchie de modules et où leurs interactions sont décrites selon un mécanisme de négociation.
- Formalisme : le système multi-agents est caractérisé par l'utilisation de moyens qui permettent la description formelle des paradigmes des agents et de leurs interactions, par exemple un système multi-agents où les agents sont décrits au moyen de la théorie de la logique et où leurs interactions sont décrites en terme de réseaux de Petri.
- Environnements : le système multi-agents est caractérisé par les outils logiciels et matériels utilisés pour développer et gérer l'exécution de ce système multi-agents, en partant en général de leurs descriptions formelles, par exemple un système multi-agents où les agents sont des objets Java, où ils communiquent entre eux par des primitives Java (par exemple RMI) et où l'exécution est assurée par une machine virtuelle Java.
- Applications : le système multi-agents est caractérisé par le domaine de problèmes auquel le système multi-agents s'applique, par exemple un système multi-agents destiné à une application commerciale, médicale, ou industrielle.

Ces quatre dimensions sont considérées comme orthogonales, dans le sens précis qu'une décision prise dans une de ces quatre dimensions n'influence pas les décisions prises dans les autres dimensions. S'il est vrai qu'ils sont orthogonaux entre eux, ils n'en ont pas pour autant une intersection vide. En effet la classification reste vague, puisqu'il peut arriver que deux systèmes multi-agents, comme le souligne Amigoni [4], puissent être classés dans deux domaines différents. Par exemple : un système entièrement construit aux moyens des outils fournis par Java et dédié au domaine médical. Autrement dit, un même système multi-agents peut fort bien appartenir à deux domaines différents, mais cela ne contredit en rien leur orthogonalité. Pour pouvoir classer un système multi-agents, il faut déterminer le domaine au-

quel le système multi-agents se rattache essentiellement. Ce but peut être atteint en déterminant le domaine qui ne peut varier, quelles que soient les variations des autres domaines. Le critère consiste donc à identifier, si cela est possible, la dimension qui ne peut subir de variation sans dénaturer le système multi-agents. Par exemple, dans le cas d'un système médical écrit sur base de la technologie Java, il est clair qu'il faut classer ce système multi-agents dans le domaine des applications, car un changement de domaine d'applications dénaturerait complètement le système ; au contraire, la modification de l'environnement, en l'implémentant dans un autre langage que Java, maintient la nature de l'application.

1.4 La notion d'Agence

1.4.1 L'Agence au sein des systèmes multi-agents

La notion d'agence est inséparable du travail de Minsky dans un essai sur l'intelligence appelé «The society of mind» [21]. L'objectif de cet essai est de proposer un nouveau paradigme pour approcher l'intelligence et les phénomènes cognitifs en général et de montrer la fécondité de ce paradigme. Ce paradigme affirme que les phénomènes cognitifs sont d'une telle complexité qu'un seul modèle ne peut arriver à les modéliser. Non seulement une pluralité de modèles est plus adéquate, mais la connexion de ces modèles constitue un élément clé pour modéliser les phénomènes cognitifs. Chaque entité, appelée agent, inclut un modèle d'un phénomène et les entités coopèrent les unes avec les autres et se concurrencent entre elles. On peut ainsi concevoir à la fois la gestion de différents modèles d'un même phénomène et de modèles portant sur des phénomènes différents. L'organisation idéale de ces agents forment ainsi la «société de l'esprit», expression qui met en évidence la conception de l'intelligence comme une connexion de modèles non significatifs en eux-mêmes. La notion d'agence a ainsi été proposée par Minsky pour désigner cette interconnexion d'agents incluant chacun un modèle d'un phénomène.

D'un point de vue technique, une agence est une machine faite de composants intelligents, appelés agents, interagissant entre eux en vue d'atteindre un but commun. Un agent est défini par sa capacité inférentielle, c'est-à-dire par sa capacité à posséder un modèle (un ensemble de faits de base) et d'être capable de manipuler ces connaissances. Il s'agit d'une définition d'un agent le long de la dimension formaliste (voir Section 1.1). Selon la dimension paradigmatique, un agent peut être un processus, comme nous le verrons dans la notion d'agence anthropique. Selon la dimension

environnementale, un agent peut être soit un ordinateur soit un robot. La granularité est supposée grosse en tant que les agents exécutent des tâches non élémentaires. Nous pouvons alors définir une agence :

Définition 1.1 - une agence est un système multi-agents particulier dans lequel chaque agent est un processus, un ordinateur ou un robot et possède une capacité inférentielle.

A noter que nous nous écartons ici de la définition donnée par Amigoni [4, p.27]. Sa définition est la suivante : une agence est un système multi-agents particulier dans lequel chaque agent est *un ordinateur ou un robot* et possède une capacité inférentielle. Cette définition, en écartant la possibilité de représenter un agent par un processus logiciel, écarte de ce fait l'agence anthropique telle que nous la décrirons (voir chapitre 2, page 22), qui est pourtant bel et bien considérée comme une agence. Amigoni ne considère dans sa définition que les dimensions environnementale et applicationnelle, alors qu'il faut prendre en compte les processus, car l'agence utilise cette dimension.

A vrai dire, la définition 1.1 devrait être considérée comme une définition pratique de l'Agence. Cette définition est liée explicitement aux dimensions orthogonales et a donc une vocation plus applicative. On trouve une autre définition dans [6], plus générale :

Définition 1.2 - une agence est une machine effectuant la coopération. Elle est constituée de composants intelligents (appelés agents) interagissant les uns avec les autres en vue d'offrir et de recevoir la collaboration des autres agents pour atteindre un but commun.

Sur base de cette nouvelle définition 1.2 et de la définition 1.1, nous obtenons une définition canonique de l'agence. Nous prenons de la définition 1.1 le lien avec les systèmes multi-agents et de la définition 1.2 la généralité de la nature des agents qui semblaient faire problème dans la définition 1.1. Nous obtenons alors une nouvelle définition :

Définition 1.3 - une agence est un système multi-agents dans lequel chaque agent possède une capacité inférentielle et interagit avec les autres agents en vue d'un but commun.

Nous considérerons cette définition comme la définition canonique d'une agence. Commentons cette définition pour mettre en évidence ses éléments importants.

1. «*système multi-agents*» : une agence doit être pensée comme une instantiation des systèmes de coopération (DES), mais où les agents ont une granularité «suffisamment» forte. Tout système ce qui n'est pas un système multi-agents et tout système multi-agents avec une granularité «faible» n'est pas une agence.
2. «*inférentielle*» : les agents du système multi-agents doivent être capables d'inférence, c'est-à-dire capables de partir d'une série de faits (un modèle) en vue d'en déduire de nouveaux. Il doit donc exister au sein des agents, un programme «non élémentaire» dont on peut considérer qu'il possède un modèle. Un système multi-agents composé d'agents réactifs par exemple n'est pas une agence. Un agent réactif est un agent fournissant une réponse en fonction des entrées, sans élaboration de ces entrées. Mataré présente ainsi des systèmes multi-agents dans lesquels les agents sont réactifs. Ils obéissent à des lois du type : «Si arrêté, revenir» ou «Si secoué, arrêter». L'approche est essentiellement de type «bottom-up», c'est-à-dire que le comportement émerge de la composition de ces agents. De plus «De tels systèmes réactifs sont limités par leur manque d'état interne ; ils sont incapables d'utiliser des représentations internes» [20]. La coopération d'agents purement réactifs n'est donc pas une agence. La capacité inférentielle, conformément aux vues de Minsky, est donc essentielle aux agents d'une agence.
3. «*un but commun*» : l'agence doit avoir un et un seul but défini *a priori*. On favorise de ce fait une approche de type «top-down», c'est-à-dire que le concepteur connaît le problème unique avant de composer l'agence. On découpe le problème pour trouver la solution plutôt que de partir d'éléments pour construire la solution. Un contre-exemple, consiste en un système multi-agents avec des agents réactifs, dans lequel le comportement non nécessairement unitaire émerge des différents agents dont la conception n'est pas directement liée à un but commun. Au contraire, dans l'agence, chaque agent est conçu en vue résoudre un seul problème global.

Cette précision au sujet de la définition nous amène à formuler deux conséquences importantes :

1. *Unitaire* : puisque l'agence est toujours conçue en fonction d'un et un seul but commun, elle ne forme, malgré ses diverses entités, qu'une et une seule machine, et non pas un ensemble d'entités poursuivant un but. Elle doit être vue comme une seule machine.
2. *Echelle* : Le caractère unitaire de l'agence implique à son tour la possibilité de la considérer comme un agent. Puisqu'elle est une machine

unique et puisqu'elle effectue en son sein des opérations inférentielles, l'agence peut être considérée à son tour comme un agent inférentiel adéquat pouvant être utilisé dans une nouvelle agence.

En conclusion, l'Agence est un concept qui met l'accent sur la coopération d'entités intelligentes pour atteindre un but à priori. Cette nature de l'Agence réside d'abord dans la coopération; elle insiste sur l'articulation d'entités intelligentes, disons des programmes effectuant des tâches considérées comme non élémentaires.

1.4.2 L'Agence comme une machine de coopération

Existe-t-il une spécificité de l'Agence en tant que machine? Répondre à cette question nous donnera un nouvel éclairage au concept d'agence. Cette classification est basée sur la nature des phénomènes qu'une machine réifie. Nous reviendrons plus tard sur la notion de réification d'une machine. Affirmons seulement pour l'instant qu'une machine représente dans la réalité un modèle d'un phénomène (voir section 4.2.2). L'homme extrait un modèle d'un phénomène et réifie ce modèle dans une machine, entité artificielle.

Machine, machine à informations, et Agence

On peut donner une classification des machines au sens général, en considérant une classification du phénomène modélisé, c'est-à-dire en classifiant les réponses données à la question : quel type de phénomènes une machine peut-elle réifier? Il s'agit donc de considérer le phénomène extérieur à modéliser. Ces différents types permettent alors une classification des machines.

- Machines du monde : ces machines réifient des modèles dont les phénomènes appartiennent au monde extérieur à l'esprit humain. Les machines du monde sont ce que nous entendons communément par machines. On peut prendre l'exemple du réfrigérateur, qui réifie un phénomène du froid. Puisque le modèle porte sur l'extérieur de l'être humain, l'intelligence humaine est exclue de ce genre de machines.
- Machines à informations : ces machines réifient des modèles dont les phénomènes appartiennent à l'intelligence humaine et de l'interaction entre le monde et l'homme. Les ordinateurs et les robots sont de telles machines.
- Machines à coopération : ces machines réifient des modèles dont les phénomènes appartiennent à la coopération entre les hommes.

On voit donc la classification des phénomènes du réel qui sous-tend une telle classification : 1) le monde externe à l'esprit humain, dont son corps fait partie 2) le rapport entre l'homme et le monde externe 3) l'intelligence humaine elle-même 4) le rapport entre l'homme et lui-même. La réification de modèles portant sur la première partition sont les machines, sur la seconde et la troisième les machines à informations, la dernière les machines à coopération.

Quel est le lien entre le concept de machine de coopération et celui d'Agence ? Ce sont en fait deux mêmes idées. Nous avons vu dans une première approche de l'Agence que la coopération en était un élément fondamental. Cette approche permet d'espérer réifier plusieurs modèles d'un même phénomène ou de phénomènes liés. On retrouve cette caractéristique coopérative dans l'agence vue comme une machine coopérative : plutôt que de considérer un seul modèle, comme l'a proposé Marvin Minsky, on peut espérer pouvoir gérer plusieurs modèles au travers de la coopération. La machine ou l'Agence est essentiellement une telle machine de coopération.

1.5 Classification des Agences

Le concept d'agence, que nous venons de présenter, est susceptible d'une classification. Cette classification est basée sur les activités animales, telles qu'elles se trouvent dans toute organisation vivante suffisamment évoluée.

- La première activité est celle de la défense du territoire. L'agence peut alors être construite pour régir le territoire d'une manière générale. A un premier niveau, elle peut être une agence géographique, c'est-à-dire qu'elle perçoit une grande échelle géographique. Pensons par exemple à une agence supervisant la météo. A un second niveau, elle peut être une agence territoriale, c'est-à-dire qu'elle perçoit une échelle comparable à celle d'un pays, comme une agence d'espionnage par exemple. A un troisième niveau, elle peut être une agence urbaine, c'est-à-dire qu'elle perçoit une échelle comparable à celle d'une ville, comme par exemple une agence dédiée à la surveillance des foules dans les villes. A un quatrième niveau, elle peut être une agence domotique, c'est-à-dire qu'elle perçoit une échelle comparable à une maison, comme par exemple une agence chargée de superviser une maison.
- Outre la défense du territoire, une des activités animales universelles est la production de l'énergie nécessaire à la vie. Parmi ces activités se trouvent la chasse ou la production de nourriture. Un exemple d'une telle agence est une agence dédiée à l'automatisation au sein d'usines

- de production.
- Ensuite l'activité de reproduction est également une activité universelle. Elle vise à assurer la survie de l'espèce d'une manière générale. Un exemple d'une telle agence est une agence implantée dans le corps et permettant, par exemple par le remplacement d'un organe défectueux, d'assurer la survie d'un individu.
- Enfin, une quatrième activité prend forme avec l'homo sapiens, celle de l'abstraction.

Cet ensemble d'activités de l'homme s'appelle le système limbic, et pour chacune de ces activités on peut mettre en évidence une classe d'agences. L'agence anthropique est celle qui va nous intéresser. Elle offre un exemple concret d'agence et du paradigme de coopération suggéré par Minsky.

1.6 L'aporie de la tour de Babel

Nous voudrions à présent présenter une difficulté relative à la construction effective d'une agence, et qui d'ailleurs l'on retrouve souvent dans la construction de systèmes multi-agents. Amigoni [4, p. 29] a proposé une solution pour cette difficulté, et cette solution est importante car elle influencera la conception de la découverte scientifique (voir section 4.2.2).

Lors de la conception d'une agence, un problème critique concerne une difficulté structurale étroitement liée à la notion d'Agence et appelée *l'aporie de la tour de Babel*. D'un côté, il est utile d'avoir des agents spécialisés et, d'un autre côté, les agents doivent être uniformes.

Les raisons qui poussent à utiliser des agents spécialisés est que les agents doivent être pensés et construits par des développeurs différents avant d'être utilisés ensemble dans une agence. Il est en effet important d'utiliser des agents spécialisés pour les raisons suivantes :

- cela permet de construire une agence capable de résoudre plusieurs problèmes.
- cela permet d'augmenter la puissance de l'agence, car on peut disposer d'agents qui effectuent des mêmes tâches selon des méthodes radicalement différentes.

Les raisons qui poussent à l'uniformité des agents proviennent du fait que les agents, pour former une agence, doivent être intégrés dans un cadre de coopération commun. En effet :

- Les agents doivent s'échanger l'information qui forme la base de la coopération.
- Les agents doivent se coordonner pour atteindre l'objectif global.

Dès lors, pour construire une agence, nous faisons face au problème d'intégrer des agents spécialisés, c'est-à-dire que nous devons faire face à l'aporie de la tour de Babel. Amigoni propose de résoudre ce problème au moyen d'une structuration particulière de chaque agent de l'agence : chaque agent est structuré en un couple de *semiagents*. Le premier semiagent est appelé *op semiagent* et son but est de fournir les fonctionnalités de l'agent, le second semiagent est appelé *co semiagent* et son but est d'effectuer la coopération. L'*op semiagent* représente la partie spécialisée de l'agent, alors que le *co semiagent* représente la partie uniforme de l'agent. D'un côté l'*op semiagent* fournit des fonctions spécialisées pour des problèmes locaux. D'un autre côté, le *co semiagent* rend l'agent capable d'être une partie active dans un processus global de coopération. Dès lors, l'architecture de chaque agent d'une agence implique la coexistence d'*op semiagent*, qui peut être différent dans chaque agent de l'agence, and le *co semiagent*, qui est uniforme pour chaque agent de l'agence.

Grâce à cette solution, il est possible de construire une agence à partir d'agents hétérogènes, qui peuvent être considérés comme des *op semiagents*, et qui peuvent être intégrés avec des *co semiagents* hétérogènes pour former une agence.

1.7 Conclusion

Notre objectif dans ce chapitre était de mettre en évidence une définition de la notion d'agence. Nous avons vu que l'agence est un système multi-agents dans lequel les agents sont non élémentaires (capables de déduire de nouveaux faits à partir de faits de base) et possèdent un objectif unique commun. L'agence doit être ainsi conçue comme une machine unitaire et qui peut donc être à son tour considéré comme un agent. La notion d'agence peut être aussi approchée comme un nouveau type de machine, une machine de coopération. Le concept d'agence est également susceptible d'une classification, sur la base d'une classification préalable des fonctions communes aux animaux.

Nous voudrions pour conclure insister sur l'intérêt de la notion d'agence. Elle prend en compte la complexité d'un phénomène à modéliser en faisant coopérer des modèles du même phénomène ou en faisant coopérer des modèles de phénomènes distincts. Outre la définition formelle et technique de l'agence, il s'agit donc d'un *nouveau paradigme* qui passe de la modélisation d'un

phénomène par un seul programme à la coopération de différents modèles ou programme. La notion plus technique d'agence tente de traduire ce paradigme. Ainsi l'interaction entre différents modèles est un élément fondamental pour approcher la complexité des phénomènes.

Au terme de cet éclaircissement de la notion d'agence, il reste bien sûr à construire des architectures concrètes pouvant illustrer ce paradigme. Nous verrons en particulier le cas d'une agence anthropique. L'autre question fondamentale est de savoir quelles applications ce nouveau paradigme peut amener. Nous verrons qu'il dessinera un nouveau rôle de la machine au sein de la découverte scientifique.

Chapitre 2

L'agence anthropique

2.1 Introduction

L'agence anthropique est le fruit du travail de Nicola Gatti et de Marco Dini [22]. L'adjectif «anthropique» souligne le fait que l'agence anthropique applique le concept d'agence au corps humain. L'objectif principal est de construire une architecture *générale* d'une agence anthropique. Cela donne un exemple concret de ce que pourrait être une agence et notre but est précisément de montrer concrètement comment la construction d'une agence est possible. Dans un premier temps, nous décrivons l'idée de l'architecture globale. Ensuite nous décrivons les différents types d'agents ainsi que les tableaux qui servent d'intermédiaire. Nous verrons que cette architecture est par nature incomplète. Le mécanisme de coordination entre les agents est en effet laissé en suspens. L'agence anthropique est par nature une agence incomplète, car le mécanisme de coordination doit pouvoir changer d'une agence à l'autre, pour des raisons qui sont principalement d'ordre épistémologique comme nous le verrons dans le chapitre 4. La troisième partie de ce chapitre s'attachera à décrire un des mécanismes de coordination possible entre les agents.

2.2 Architecture générale

Dans cette section nous essayerons de donner un aperçu général de l'architecture, d'abord du point de la vue de la genèse, et ensuite à un niveau plus descriptif.

2.2.1 Développement de l'architecture

Formulons à présent le projet de l'agence anthropique. L'idée première est d'appliquer le concept d'agence au corps humain. Il s'agit de construire un système de contrôle du corps humain utilisant le concept d'agence, et par conséquent utilisant la capacité de faire coopérer différents modèles d'un même phénomène.

L'agence anthropique comme simulation du pancréas

Prenons d'emblée un exemple : on pourrait souhaiter palier au dysfonctionnement d'un pancréas. Pour cela on implanterait dans le corps une machine, une particularisation de l'architecture de l'agence anthropique, pourvue de capteurs et d'actuateurs. L'agence anthropique construite aurait bien entendu une capacité d'élaboration la rendant apte à prendre des décisions. D'abord elle vérifierait l'absorption de nourriture du corps et ajusterait en conséquence le taux d'insuline dans le sang. La prise en compte du taux de nourriture dans le sang constituerait un modèle particulier. Mais l'agence anthropique, conformément au concept d'agence, devrait être capable de faire coopérer différents modèles du même phénomène, en l'occurrence le pancréas. Un second modèle pourrait cohabiter avec le premier. Ce second modèle serait, par exemple, une détermination du taux glucose en fonction du taux de stress de l'individu porteur de l'agence anthropique. Deux modèles ou plus cohabiteraient au sein de l'agence anthropique, qui rendrait effective la capacité de multi-modélisation de l'agence dans le cadre d'un système de contrôle du corps humain.

Problème de la conception de l'architecture

Le problème de la conception de l'agence anthropique se présente comme la recherche d'une architecture :

- dont l'objectif est un *système de contrôle* d'une partie du corps humain. C'est le pôle proprement anthropique de cette agence.
- *générique*, c'est-à-dire potentiellement utilisable pour différentes parties du corps
- capable de réaliser la *capacité de multi-modélisation* entre différents modèles. C'est le pôle agence.

- *modulable*, c'est-à-dire que des entités (des agents) peuvent être ajoutés dynamiquement au sein de l'agence anthropique. On imagine facilement l'ajout dynamique d'un nouvel agent incluant un modèle supplémentaire.

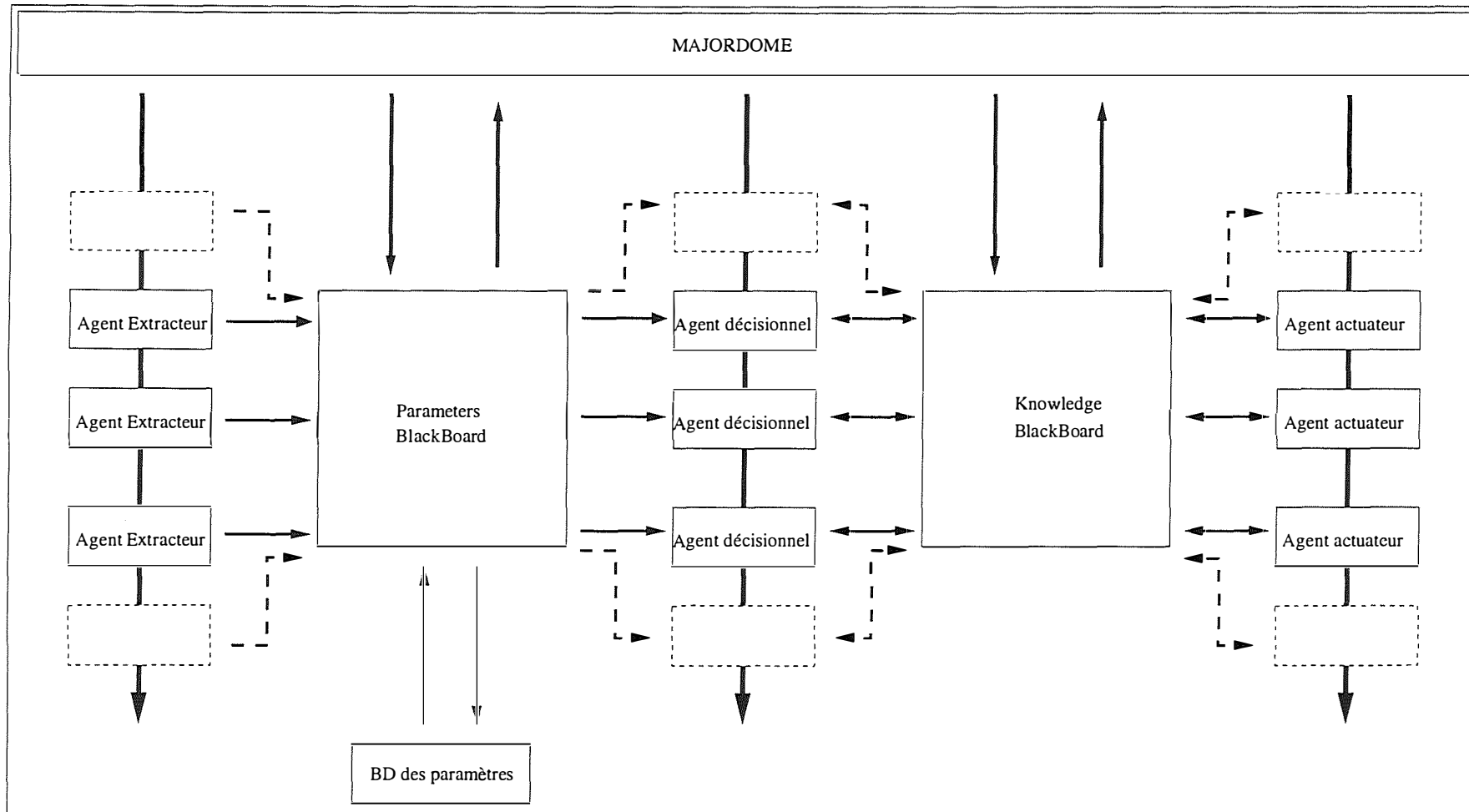


FIG. 2.1 – L'architecture logique de l'agence anthropique. Les capteurs et les actuateurs ne sont pas représentés.

Développement de l'architecture

Le développement de l'architecture est pensé selon trois axes :

1. Capacité de perception.

La première fonctionnalité nécessaire pour un système de contrôle est la capacité de prélever des informations sur l'environnement, dans notre cas précis le corps humain. Cette capacité de perception doit être générique et modulable. D'abord la capacité de perception doit être générique, laissant la possibilité à l'utilisateur d'inclure différents types de capteurs conformément à l'application particulière choisie. Ensuite la capacité de perception doit être modulable, c'est-à-dire qu'il doit être possible d'ajouter dynamiquement de nouveaux senseurs durant le fonctionnement de l'agence anthropique.

Ces deux objectifs sont atteints par la présence de capteurs dans le corps et par la création d'un agent logiciel appelée agent extracteur. La lecture des signaux est assurée par la présence de capteurs. L'agence anthropique possède un **réseau de capteurs**, éventuellement distribué, qui capte des grandeurs physiques et les transforme en signaux. Cette partie est une partie matérielle, qui comme telle n'est pas modifiable. On définit au préalable les besoins de perception et on inclut dans le réseau de capteur les éléments matériels nécessaires. Un **agent extracteur** a pour rôle d'extraire des signaux le ou les paramètre(s) utiles aux modèles de l'agence. Chaque agent extracteur est donc connecté au réseau de capteurs. Cet agent extracteur est logiciel. Un agent extracteur a donc en entrée un ensemble de signaux et en sortie un ensemble de paramètres. A un signal peuvent correspondre un ou plusieurs paramètres, et à plusieurs signaux peuvent correspondre un ou plusieurs paramètres.

La figure 2.1 montre l'ensemble des agents extracteurs effectuant le travail de perception au sein de l'agence anthropique. L'exigence de généricité est atteinte par l'indétermination du réseau des capteurs ; l'exigence de modularité est atteinte par l'utilisation d'agents qui pourront s'inscrire au sein de l'agence anthropique.

2. Capacité d'élaboration.

La seconde fonctionnalité nécessaire pour un système de contrôle est la capacité d'élaboration, rendant l'agence anthropique capable de prendre une décision. Puisque c'est à ce niveau que se situe les différents modèles et que se prend la décision d'actuation de l'agence anthropique, on retrouve deux types d'exigences :

- (a) D'abord ce niveau doit rendre possible la capacité de multi-modélisation, permettant à l'agence d'articuler différents modèles.
- (b) Ensuite l'exigence de modularité est également cruciale, puisque l'agence doit être capable d'incorporer dynamiquement de nouveaux modèles.

Remarquons que la modularité des modèles impliquent nécessairement la modularité de la perception et de l'actuation : il se peut qu'un modèle nouvellement inséré ait besoin d'un agent extracteur et d'une capacité d'actuation qui ne sont pas encore gérés logiciellement par l'agence. Dans ce cas, il doit également être possible d'inscrire dynamiquement un nouvel agent extracteur et une nouvelle capacité d'actuation.

Comment atteindre ces deux objectifs ? Ils sont atteints par la présence d'agents décisionnels, chacun d'eux incluant un modèle particulier. Un **agent décisionnel** a pour fonction de décider à quel moment et de quelle manière agir. Un agent décisionnel est donc un agent capable de prendre une décision quant au nouvel état à atteindre. Il possède en entrée un ensemble de paramètres (préalablement obtenus par les agents extracteurs) et un modèle. Ce modèle est généralement un modèle *partiel* du phénomène. Il possède également des informations sur les capacités d'actuation de l'agence puisqu'il doit décider de l'action à entreprendre. La figure 2.1 montre l'ensemble des agents décisionnels.

Le mécanisme de coordination entre les différents modèles serait trop complexe à exposer ici ; la section 2.4 est consacrée à ce problème.

3. Capacité d'actuation.

La troisième fonctionnalité nécessaire d'un système de contrôle du corps est la capacité d'actuation. Symétriquement au réseau de capteurs et aux agents extracteurs, cette troisième fonctionnalité se compose d'un **réseau d'actuateur** et d'entités logicielles appelées **agents actuateurs**. Les objectifs de généricité et de modularité sont atteints de la même manière que pour les agents extracteurs. La figure 2.1 montre l'ensemble des agents extracteurs.

4. Communication entre les agents extracteurs et les agents décisionnels.

Comment, à présent, faire communiquer les agents extracteurs, qui obtiennent les paramètres des signaux tirés par le réseau de capteurs, et les agents décisionnels, qui utilisent ces paramètres pour communiquer une décision aux agents actuateurs ? Il faut garder à l'esprit l'exigence de modularité. Un mécanisme de communication complexe entre les agents extracteurs et décisionnels serait un obstacle à la modularité.

Une solution naturelle consisterait à ce que chaque agent nouvellement inséré contacte chaque agent extracteur pour lui demander si cet agent extracteur le paramètre qu'il désire. Mais cette solution doit être écartée pour trois raisons :

- (a) Elle sacrifie la modularité. Supposons que deux agents extracteurs produisent le paramètre recherché par un agent décisionnel, l'un des agents extracteurs ne produisant que ce paramètre, l'autre agent extracteur produisant plusieurs paramètres. Supposons également que l'agent décisionnel considéré ne contacte que le premier de ces agents décisionnels. Si le premier agent décisionnel est supprimé, nous devrions, si la modularité est parfaite, ne pas porter atteinte au bon fonctionnement de l'agence. Or, avec cette solution, on voit que l'agent décisionnel n'aurait plus accès au paramètre, à moins qu'il ne recontacte l'ensemble des agents extracteurs, dans le cas général où un autre agent extracteur aurait été changé pour prélevé le paramètre.
- (b) Bien que ces mécanismes ne soient pas impossibles à mettre en place (on pourrait imaginer que l'agent décisionnel, lorsqu'il ne peut plus contacter un agent extracteur, recontacte l'ensemble des agents extracteurs disponibles à ce moment), ils sont néanmoins complexes à mettre en oeuvre.
- (c) Ces mécanismes bloquent les agents extracteurs et les agents décisionnels dans leur fonctionnement puisque ceux-ci doivent gérer l'ensemble des communications.

Une seconde solution, qui sera celle adoptée dans l'architecture de l'agence, éliminera ces deux problèmes, à savoir la complexité des mécanismes de communication à mettre en oeuvre et la charge supportée par les agents lors de la communication. On utilise un tableau intermédiaire, sur lequel les agents extracteurs s'inscrivent en indiquant le paramètre qu'ils sont capables de prélever. Les agents extracteurs inscrivent alors périodiquement la valeur du paramètre qu'il échantillonne. Si plusieurs agents extracteurs échantillonnent le même paramètre, ces agents extracteurs écrasent la valeur qui se trouve sur le tableau. Les agents décisionnel s'inscrivent également à ce tableau et lisent la valeur du paramètre qui les intéresse.

De cette manière, nous évitons les problèmes évoqués dans la solution initiale. Les mécanismes de communication sont simples (écriture pour les agents extracteurs, lecture pour les agents décisionnels), la modularité est accrue puisque l'insertion dynamique d'un agent est simplifiée,

et aucun des agents n'est bloqué puisque la lecture ou l'écriture est immédiate.

Ce tableau de premier niveau est appelé le *Parameter BlackBoard* (PBB). La figure 2.1 montre la médiation du Parameter BlackBoard entre les agents extracteurs et les agents décisionnels. Le Parameter BlackBoard possède également une base de données dans laquelle il est capable d'enregistrer divers paramètres qui peuvent se révéler intéressants à exploiter pour le corps médical.

5. Communication entre les agents actuateurs et les agents décisionnels.

Comment, à présent, faire communiquer les agents décisionnels et les agents actuateurs ? Ce problème cache en fait un triple problème, et ne relève pas de la simple communication de valeur comme dans le cas du premier niveau.

- (a) Il faut tout d'abord faire communiquer les agents décisionnels entre eux. Cette communication a pour but d'établir un objectif commun aux agents décisionnels.
- (b) Il faut ensuite que les agents actuateurs rendent compte des possibilités d'actuation aux agents décisionnels, de sorte que ceux-ci recherchent un objectif réalisable par les moyens d'actuation actuellement disponibles dans l'agence anthropique.
- (c) Il faut enfin, une fois cet objectif décidé par les agents décisionnels, répartir le plus efficacement possible (c'est-à-dire avec le moins de désagrément possible pour le corps) les actions à réaliser aux agents actuateurs.

Ce triple problème, s'il reste sans réponse, nous donne une architecture incomplète, c'est-à-dire une architecture ne possédant pas de mécanisme de coordination. Le problème de la coordination au sein de l'agence anthropique, qui est le problème central de l'agence anthropique, serait beaucoup trop long à développer ici. La section 2.4 lui est consacrée. Notons simplement ici que la solution retenue sera également l'utilisation d'un tableau de second niveau, nommé le *Knowledge BlackBoard* (KBB).

6. Communication vers l'extérieur.

L'agence anthropique possède enfin un moyen de communication vers l'extérieur. Il s'agit du **Majordome** (voir figure 2.1) qui permettra à un spécialiste ainsi qu'à l'utilisateur de vérifier le bon fonctionnement de l'agence, de choisir certains paramètres, d'ajouter un agent, d'ajouter

un modèle en ajoutant un agent décisionnel, de supprimer un modèle, de consulter la base de données des paramètres, etc.

2.2.2 Description de l'architecture

Décrivons à présent l'architecture logique de l'agence anthropique, principalement les différents types d'agents et les unités de communication existantes. La figure 2.1 montre l'architecture logique de l'agence anthropique, où nous avons exclu les éléments matériels.

- **lecture des signaux** : la présence de capteurs est importante pour pouvoir percevoir l'environnement. L'agence anthropique possède un réseau de capteurs, éventuellement distribué, qui capte des grandeurs physiques et les transforme en signaux. Cette partie est une partie matérielle, qui comme telle n'est pas modifiable.

- **agent extracteur** : cet agent a pour rôle d'extraire des signaux le ou les paramètre(s) utiles aux modèles de l'agence. Un agent extracteur a donc en entrée un ensemble de signaux et en sortie un ensemble de paramètres. A un signal peuvent correspondre un ou plusieurs paramètres, et à plusieurs signaux peuvent correspondre un ou plusieurs paramètres.

Un agent extracteur doit être capable de mettre à jour son algorithme, de fournir ses paramètres à l'agence, et doit pouvoir être suspendu si le paramètre n'est plus utilisé par l'agence. Certains paramètres de fonctionnement de l'agent extracteur doivent également être accessibles, par exemple la fréquence d'extraction.

- **Parameter BlackBoard** : sa fonction principale au sein de l'agence est de récolter les paramètres extraits par les agents extracteurs et de les rendre disponibles aux autres agents. Il s'agit donc d'un tableau sur lequel sont écrits et lus les paramètres.

On nomme ce tableau *tableau de premier niveau* ou *Parameter BlackBoard (PBB)*.

- **Base de données des paramètres** : sa fonction consiste à enregistrer les données relevées par l'agence. Ces données pourraient ainsi être utilisées dans le champ de la recherche médicale. La base de données devrait posséder divers moyen de filtrage des données (data-mining par exemple) et, pour assurer la modularité, devrait être informée de l'apparition d'un nouveau paramètre et d'en commencer l'archivage.

- **agent analyseur** : sa fonction est de donner une valeur heuristique, sur base de paramètres et de son modèle, estimant le bon fonctionnement d'une partie du corps humain. Il surveille donc une partie du corps humain en confrontant les paramètres avec son modèle.
La fonction de cet agent s'apparente donc à celle d'un expert, et son modèle s'apparente à la connaissance de cet expert.
- **agent abstrait** : sa fonction est d'indiquer la présence d'un état. Par exemple, l'agent abstrait pourrait détecter la présence d'activité physique. La sortie de l'agent est donc une variable booléenne indiquant la présence de l'état. Il ne s'agit plus, comme dans le cas de l'agent analyseur, de confronter des paramètres avec un modèle.
- **agent décisionnel** : sa fonction est de décider à quel moment et de quelle manière agir. Un agent décisionnel est donc un agent analyseur qui est capable de prendre une décision quant au nouvel état à atteindre. Il possède en entrée un ensemble de paramètres (lus sur le PBB) et un modèle. Ce modèle est généralement un modèle *partiel* du phénomène. Il possède également des informations sur les capacités d'actuation de l'agence puisqu'il doit décider de l'action à entreprendre. Il est important de noter que plusieurs agents décisionnels peuvent prendre des décisions contradictoires au sujet du même paramètre. On parle alors d'une situation de conflit, qui déclenche un mécanisme de coordination complexe que nous détaillerons dans la section 2.4.
- **agent actuateur** : sa fonction est de commander l'actuateur physique et de faire connaître les possibilités réelles d'actuation. L'agent actuateur doit pouvoir informer l'agence (principalement les agents décisionnels) sur les possibilités d'actuation. Par exemple, il est possible que l'on ne puisse pas diminuer une substance mais seulement l'augmenter, ou vice versa. De plus, l'agent actuateur devra informer sur la difficulté d'effectuer une action.
- **Knowledge BlackBoard** : sa fonction est de servir, comme le PBB, de moyen de communication bi-directionnelle entre les agents décisionnels et les agents actuateurs ainsi que de gérer tout conflit le cas échéant. On l'appelle *tableau de second niveau* ou *Knowledge BlackBoard (KBB)*.
- **majordome** : sa fonction principale est de servir de pont vers l'extérieur. D'abord le majordome rend l'ensemble des informations de

l'agence disponible et donne ainsi une vue d'ensemble du fonctionnement de l'agence. Ensuite il offre des fonctions d'ajournement, comme celle d'ajouter et de retirer un agent ou celle de mettre à jour les algorithmes des agents extracteurs ou les modèles des agents décisionnels. Le majordome communique exclusivement au travers du PBB et du KBB et gère l'agence au travers de ces deux tableaux. Par exemple, les agents décisionnels donneront des références à leur propre fonction d'ajournement de modèle au PBB, et le majordome en prendra connaissance via ce tableau de premier niveau.

2.3 Les agents, les tableaux et le majordome

Nous présentons dans cette section les éléments essentiels de l'agence dans le détail.

2.3.1 Agent extracteur

L'agent extracteur (AE) a pour mission de transformer les signaux acquis par les capteurs en paramètres et d'écrire ces paramètres sur le PBB pour le rendre disponible aux agents décisionnels. La figure 2.2 montre le schéma logique d'un agent extracteur.

L'extraction des signaux et calcul des paramètres

La première étape pour l'agent d'extraction est l'extraction des signaux qui s'effectue par la lecture des capteurs selon une fréquence d'échantillonnage.

La seconde étape est le calcul effectif des paramètres. Rappelons, comme indiqué à la section 2.2.2, que le lien entre signaux et paramètres est variable : deux signaux peuvent servir pour formuler un paramètre, un signal pour un seul paramètre, etc. Ceci est représenté par les différentes flèches entre les échantillonneurs et les algorithmes dans la figure 2.2.

Les paramètres sont ensuite communiqués au PBB.

Intégration d'un nouvel agent extracteur

Avant de passer les paramètres au PBB, un agent extracteur aura dû s'identifier au sein de l'agence et déclarer les paramètres qu'il communique à l'agence.

L'*intégration* d'un agent est ce processus par lequel l'agent s'insère et devient opérationnel au sein de l'agence.

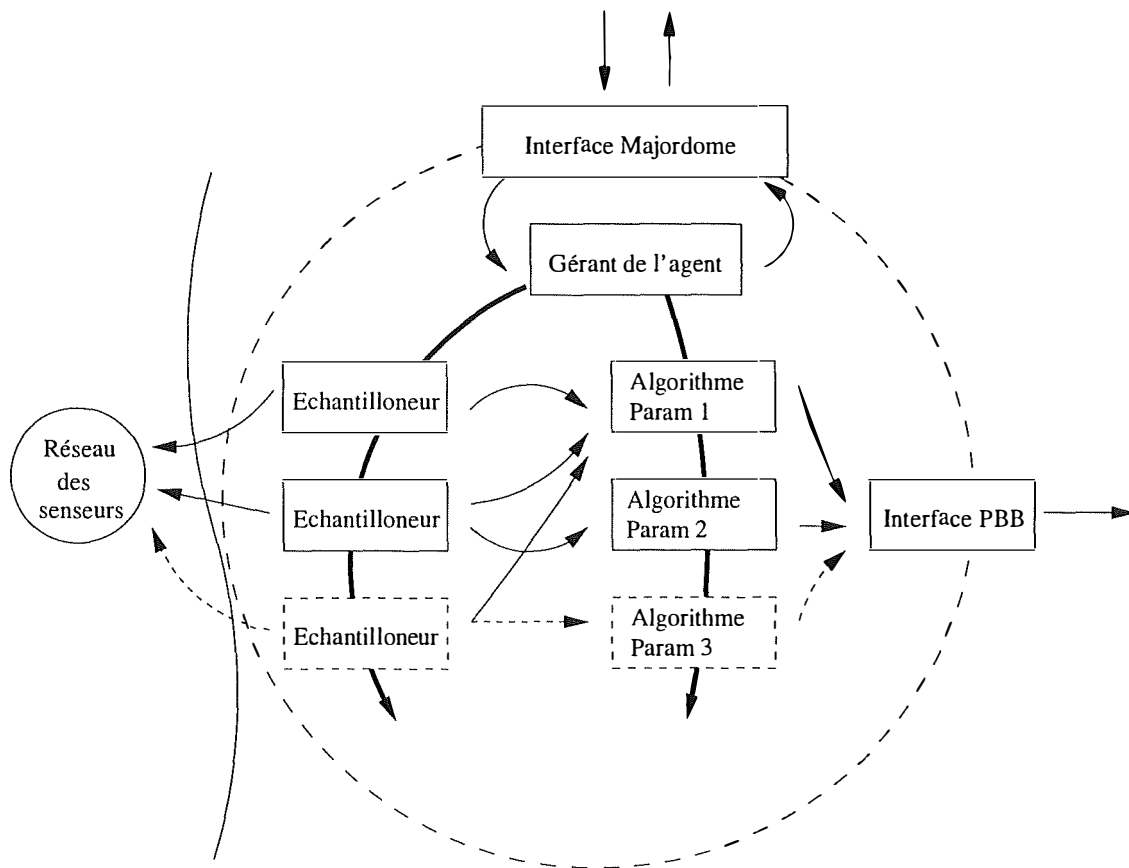


FIG. 2.2 – Schéma logique d'un agent extracteur.

L'intégration d'un agent extracteur est médiatisé par la connexion au réseau de capteurs et au PBB. On supposera que l'agent connaît le moyen (l'adresse réseau par exemple) de joindre et de lire les capteurs matériels. Nous ne nous occupons ici que de la connexion au PBB.

L'agent extracteur se connecte au serveur du PBB et demande un nouveau numéro d'identifiant. L'intégration se produit par la mémorisation, pour l'agent, de son numéro d'identifiant et pour le PBB, d'un lien entre le numéro d'identifiant et une référence réseau à l'agent extracteur. Plus précisément :

1. AE : accès au serveur du PBB
2. AE : demande d'un nouveau numéro d'identifiant au PBB
3. PBB : recherche d'un identifiant libre
4. PBB : renvoie de ce numéro d'identifiant
5. AE : réception et mémorisation de ce numéro

6. AE : envoie sa référence à lui-même
7. PBB : mémorise l'association (identifiant, référence à cet AE).

Notons à ce point qu'en donnant une référence à lui-même au PBB l'agent extracteur fournit en même temps un ensemble de services accessible au majordome et l'ensemble des informations le concernant.

Une fois que l'agent extracteur est identifié, il doit encore faire enregistrer au PBB les paramètres qu'il produit. Ce mécanisme s'effectue pour chaque paramètre de manière similaire :

1. AE : demande d'un nouveau numéro d'identifiant au PBB pour le paramètre
3. PBB : recherche d'un identifiant disponible dans l'ensemble des identifiants pour les paramètres.
4. PBB : renvoie de ce numéro d'identifiant
5. AE : réception et mémorisation de ce numéro pour le paramètre
6. AE : envoie les caractéristiques du paramètre
7. PBB : mémorise l'association (identifiant, référence au paramètre) pour cet AE.

Au terme de ce processus d'intégration, l'agent extracteur est prêt à communiquer son ou ses paramètre(s) au PBB.

Passage de paramètres

L'agent extracteur extrait des capteurs les signaux selon une certaine fréquence d'échantillonnage, calcule le ou les paramètre(s) et les communique au PBB.

Le PBB possède ainsi continuellement la dernière valeur d'un paramètre communiquée et la rend disponible aux agents décisionnels. De cette manière le système est plus robuste : il n'y a pas de liens complexes entre agents extracteurs et décisionnels. Les agents extracteurs écrivent selon leur fréquence propre sur le PBB et les agents décisionnels lisent les valeurs lorsqu'ils en ont besoin sur le PBB.

Notons qu'il est souhaitable que la fréquence de lecture des agents décisionnels soient supérieurs ou égal à la fréquence d'extraction des agents extracteurs. En effet si la fréquence de lecture d'un agent décisionnel est trop faible, ses décisions ne pourraient plus correspondre à l'état réel de l'environnement.

Interaction avec le majordome

Le majordome doit rendre public la liste des agents extracteurs et pour chacun l'ensemble des informations les concernant. Il doit également rendre public, pour chaque extracteur, un certain nombre de services de modification dynamique.

Le majordome obtient ces informations au travers du PBB. Il questionne le PBB sur la liste des agents extracteurs, les paramètres qui leur sont associés, et les fonctions publiques de ces agents. Le Majordome communique seulement par le PBB (et le KBB) comme déjà indiqué dans la figure 2.1. Dans la figure 2.2, nous considérons d'un point de vue logique que le majordome est en lien direct avec l'agent, même si du point de vue de l'implémentation la médiation du PBB est requise.

Le *gérant* de l'agent extracteur est chargé d'exécuter les fonctions publiques qui seront demandées à l'agent extracteur.

Parmi ces fonctions, on compte :

1. ajout ou élimination d'un algorithme d'extraction
2. ajournement d'un algorithme d'extraction
3. interruption de l'activité d'extraction d'un paramètre
4. variation de la fréquence d'extraction d'un paramètre.
5. variation de la fréquence d'échantillonnage d'un signal.

2.3.2 Agent décisionnel

Les agents décisionnels (AD) constituent le coeur de l'agence anthropique. C'est à eux qu'échoit le rôle de déterminer, sur base de leur propre modèle, à quel moment l'agence anthropique doit agir et comment elle doit agir. Dans cette section nous allons détailler le fonctionnement des agents décisionnels (voir figure 2.3) sans rentrer dans le détail du mécanisme de coordination entre les agents que nous aborderons à la section 2.4.

Intégration d'un agent décisionnel

L'intégration d'un agent décisionnel s'effectue par une connexion au PBB et au KBB. D'abord l'agent décisionnel s'enregistre dans le KBB :

1. AD : accès au serveur du KBB
2. AD : demande d'un nouveau numéro d'identifiant au KBB
3. KBB : recherche d'un identifiant libre, univoque pour toute l'agence
4. KBB : renvoi de ce numéro d'identifiant

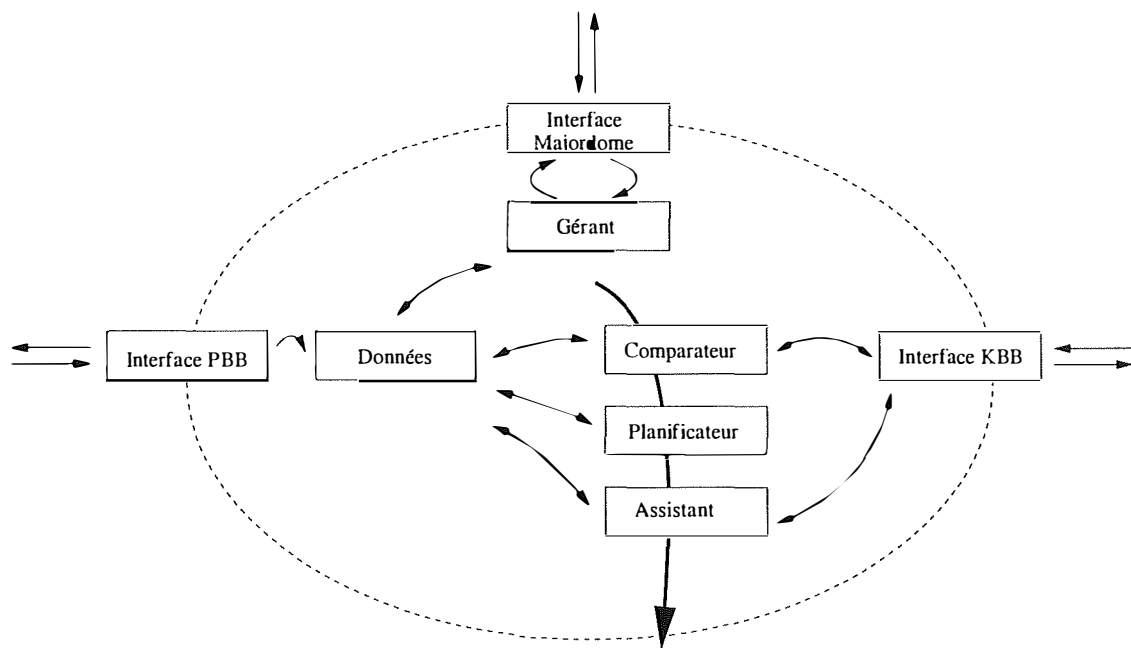


FIG. 2.3 – Schéma logique d'un agent décisionnel.

5. AD : réception et mémorisation de ce numéro
6. AD : envoi au KBB d'une référence à lui-même
7. KBB : mémorise l'association (identifiant, référence à cet AD).

L'agent décisionnel s'enregistre ensuite au PBB. L'agent décisionnel enregistre enfin au PBB chaque paramètre qu'il a besoin de lire :

1. AD : demande d'un paramètre
2. PBB : recherche de ce paramètre et création de l'association (identifiant de l'AG, identifiant du paramètre)
3. PBB : renvoi de l'identifiant du paramètre
4. AD : mémorise l'identifiant du paramètre.

L'agent décisionnel possède ainsi l'identifiant des paramètres qu'il est alors capable de lire sur le PBB. L'agent décisionnel a encore besoin de connaître les paramètres sur lesquels l'agence anthropique peut agir, c'est-à-dire les paramètres effectivement gérés par les agents actuateurs; il a aussi besoin de savoir pour chaque paramètre les possibilités exactes d'actuation.

1. AD : demande de la liste des paramètres gérés par les agents actuateurs

au KBB et des possibilités d'actuation sur ces paramètres.

2. KBB : renvoi d'une liste de couples (paramètre, possibilités d'actuation).

A ce stade, il peut dès lors commencer à évaluer l'état de bien-être du système qu'il surveille grâce à son modèle partiel, et éventuellement décider d'une action si l'état n'est plus satisfaisant.

Représentation interne du modèle

Un agent décisionnel possède un modèle partiel du phénomène que l'agence anthropique surveille. Sur base des paramètres qu'il lit dans le KBB, il évalue grâce au modèle si les paramètres prélevés sont admissibles, c'est-à-dire s'ils sont non pathologiques. Nous allons dans ce paragraphe décrire le formalisme du modèle tel qu'il se trouve au sein d'un agent décisionnel.

Un modèle \mathbf{m} à n paramètres d'un phénomène \mathbf{p} dans un agent décisionnel est un espace multi-dimensionnel à n dimensions, chaque dimension représentant les valeurs possibles d'un paramètre. Chaque point (c_1, c_2, \dots, c_n) , correspondant à un état physique, possède une valeur numérique entière au sein du modèle. Plus cette valeur est grande, plus l'état est considéré par l'agent comme pathologique. Une valeur minimale indique un état optimal.

Mathématiquement, un modèle est donc une fonction :

$$Mod : \mathcal{R}^n \rightarrow [min, max] \quad \forall \quad min, max \in \mathbb{Z}$$

La valeur exacte de min et de max est arbitraire et est fixée, respectivement, à 0 (état sain) et 100 (état pathologique).

De plus, chaque dimension, associée à un paramètre, possède une échelle découpant la dimension en k intervalles $[i_0, i_1], [i_1, i_2], \dots, [i_{k-1}, i_k]$. L'utilisation d'une telle échelle se justifie d'une part par l'impossibilité de représenter tous les points de l'intervalle (principalement pour des raisons de mémoire), et d'autre part par la constatation que certaines zones ont une valeur uniforme. Généralement seuls certains intervalles au sein de la dimension auront des valeurs différentes de la valeur maximale. Ces intervalles constituent la zone significative pour ce paramètre. On adoptera un découpage de la dimension plus fin autour des zones significatives et un découpage plus grossier dans les zones ne portant pas d'information.

Ainsi, si l'on prend un agent surveillant le taux de glucose et le taux d'insuline, le modèle se compose d'un espace à deux dimensions dans lequel les points correspondant aux états pathologiques ont une grande valeur, et

les points correspondant à un état optimal une valeur minimale. Les autres points ont des valeurs intermédiaires. Lorsqu'il lit les paramètres sur le KBB, l'agent s'assure que l'état courant est non pathologique, c'est-à-dire qu'à cet état correspond dans le modèle de l'agent une valeur faible (i.e. proche de zéro).

Capacité de multi-modélisation

Le modèle de chaque agent décisionnel est *partiel*. Cela signifie que, si l'agence anthropique vise à remplacer un phénomène p du corps, divers agents décisionnels possèdent des modèles différents du modèle p . Ces modèles partiels peuvent ne pas avoir d'intersection entre eux, peuvent avoir une dimension en commun, ou toutes les dimensions en commun. Ce dernier cas évoque la capacité de faire coopérer des modèles alternatifs à un même phénomène.

Prenons, pour éclaircir ce point un exemple concret. Soit une agence anthropique modélisant le phénomène du pancréas, c'est-à-dire essayant de contrôler le taux de glucose dans le système métabolique. Un premier agent peut posséder un modèle ayant pour dimensions le glucose et le taux de nourriture présents dans le système métabolique. Un second agent décisionnel pourrait posséder un modèle ayant pour dimensions le glucose d'une part et le taux d'activité physique d'autre part. Un troisième agent décisionnel pourrait posséder un modèle ayant pour dimensions le glucose d'une part et le taux de stress d'autre part. Enfin un quatrième agent pourrait posséder un modèle alternatif au premier, ayant pour dimensions également le glucose et le taux de nourriture.

L'agence anthropique est capable d'accepter dynamiquement un nouvel agent décisionnel, par le processus d'intégration de ce nouvel agent. Un modèle peut alors être ajouté, et s'il possède une intersection avec les autres modèles de l'agence anthropique, permet par là de faire coopérer ces différents modèles.

Ainsi émerge la propriété de *multi-modélisation* qui est la capacité d'un système de faire coexister des modèles partiels et/ou alternatifs en vue d'émuler un phénomène.

Évaluation de l'état de bien-être

Lorsque l'agent décisionnel a lu les valeurs des paramètres dans le PBB, il doit décider s'il doit entreprendre une action, autrement dit il doit évaluer si l'état est bon.

C'est justement le rôle du *comparateur* (voir figure 2.3). Celui-ci possède un paramètre appelé *seuil d'état de bien-être* qui est une valeur comprise

entre $[min, max]$. Notons ce seuil s . Le comparateur extrait d'abord la valeur de l'état courant, $Mod(c_1, c_2, c_3, \dots, c_n)$, et si $Mod(c_1, c_2, c_3, \dots, c_n) > (max - s)$ décide que l'état courant n'est plus satisfaisant, et active alors le planificateur.

Un cas particulier doit être considéré si une action avait déjà été demandée par l'agent décisionnel. Si $Mod(p_1, p_2, \dots, p_n) \leq (max - s)$, le comparateur se met simplement en attente d'une nouvelle valeur des états physiques (c_1, \dots, c_n) . Le cas contraire entraîne un nouveau comportement du comparateur.

Si une ou plusieurs actions ont déjà été demandées par l'agent décisionnel, l'agent décisionnel évalue la différence à un instant t entre la valeur attendue $v_a(t)$ et la valeur réelle $v_r(t)$:

$$E(t) = \frac{|v_a(t) - v_r(t)|}{\left(\frac{v_a(t) + v_r(t)}{2}\right)} * 100$$

Nous appellerons $E(t)$ *erreur pourcentuelle*. L'agent décisionnel confronte alors la valeur $E(t)$ avec le *seuil d'erreur* E . Si

$$E(t) > E$$

alors le comparateur planifiera une nouvelle action en se basant seulement sur la différence entre $v_a(t)$ et $v_r(t)$.

Pensons par exemple à une personne qui se met à faire de l'activité physique. Supposons que l'agence anthropique, avant cette nouvelle activité, avait décidé d'une action adéquate. Or à la prochaine lecture sur le PBB, l'agent décisionnel se rend compte que l'action n'a pas l'effet désiré ; dans ce cas, l'agent décisionnel utilise, comme nous venons de le voir, une méthode permettant d'atteindre la situation souhaitée.

Prise de décision des agents

Si un agent a décidé que l'état actuel était inférieur au seuil d'état de bien-être, le comparateur active le planificateur.

Le rôle du planificateur est de rechercher plusieurs points de coordonnées $(p_1, p_2, p_3, \dots, p_n)$, au sein du modèle à partir de la position courante (c_1, \dots, c_n) communiquée par le PBB, qui satisfont la contrainte $Mod(p_1, \dots, p_n) \leq (max - s)$. Le planificateur pratique une recherche heuristique au sein du modèle en vue d'obtenir de tels points. Ces points correspondent à un état jugé sain. Sans rentrer dans le détail technique d'une telle recherche, l'algorithme tend à suivre les chemins qui descendent, c'est-à-dire vers les points voisins qui ont une valeur inférieure. Ils essaient également de favoriser la découverte d'un point tel que $Mod(p_1, \dots, p_n) = min$ (minimum global).

A ce stade le planificateur donne les points à atteindre à l'assistant qui amorce un mécanisme complexe de coopération avec les autres agents décisionnels par la médiation du KBB. Nous détaillerons le mécanisme fonctionnement de l'assistant lors de la description du mécanisme de coordination dans la section 2.4.

Interaction avec le Majordome

Le majordome interagit, comme dans le cas de l'agent extracteur, d'un point de vue logique directement avec l'agent décisionnel (voir figure 2.3) et du point de vue de l'implémentation le majordome interagit en fait avec l'agent décisionnel au travers du KBB et du PBB.

La liste des agents décisionnels et l'ensemble des informations sur ces agents peuvent être obtenus directement au KBB. La liste des paramètres que l'agent décisionnel utilise est disponible dans le PBB.

Enfin le majordome a accès aux fonctions publiques de chaque agent décisionnel par l'intermédiaire du KBB. Parmi ces fonctions, notons :

- choix du seuil de bien-être
- choix du seuil d'erreur pourcentuelle
- ajournement du modèle de l'agent décisionnel
- interruption de l'activité de planification
- variation de la fréquence de prélèvement d'un paramètre

Le gérant a pour fonction d'exécuter ces différentes fonctions au sein de l'agent décisionnel.

2.3.3 Agent actuateur

Dans cette section nous détaillerons la description d'un agent actuateur. Rappelons que, comme l'agent extracteur, il est relié à un ensemble d'actuateurs matériels avec lesquels il est capable de communiquer. Son rôle, outre le contrôle des actuateurs, est d'informer l'agence anthropique sur les possibilités d'actuation des actuateurs matériels et de la difficulté d'entreprendre certaines actions. La figure 2.4 montre la structure d'un agent actuateur.

Spécification de l'actuation

L'agence anthropique, comme système de contrôle, vise à surveiller le bon fonctionnement d'une partie du corps. L'agence anthropique doit donc être capable d'évaluer l'effet des actions qu'elle entreprend au sein du corps. Cette

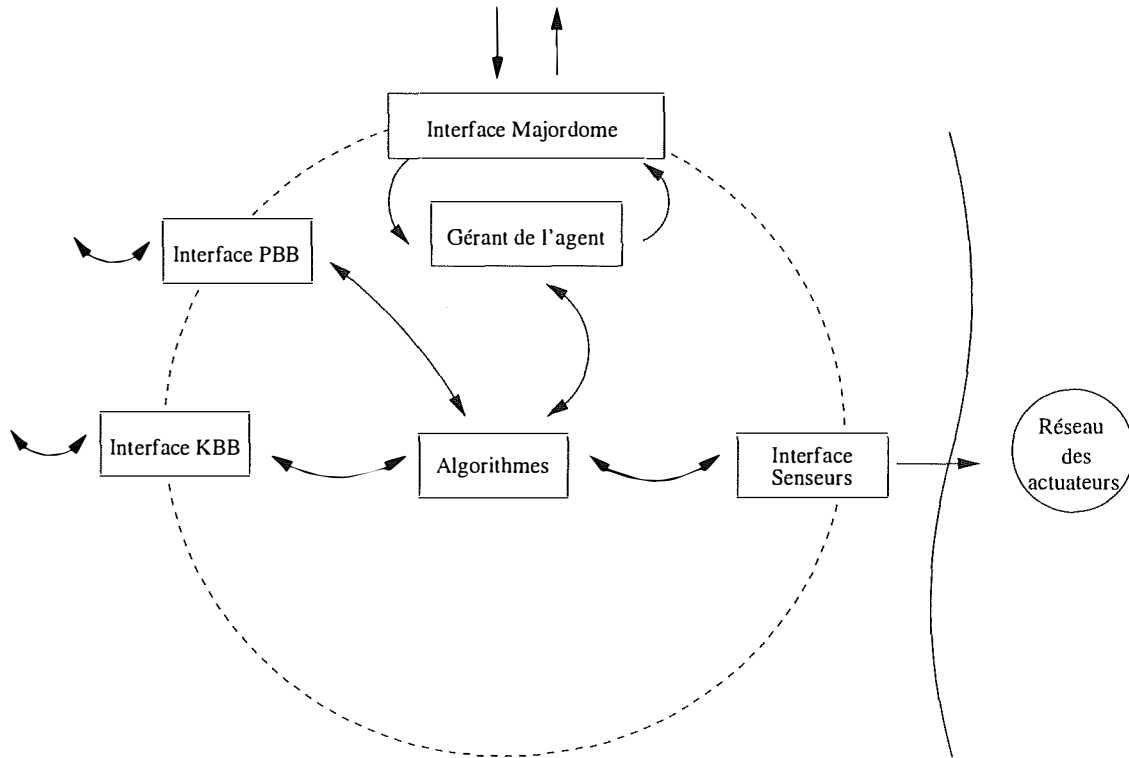


FIG. 2.4 – Schéma logique d'un agent actuateur.

première exigence formule qu'à tout paramètre contrôlé P_c l'agence anthropique doit pourvoir lire son correspondant P_e à travers les agents extracteurs :

$$\bigcup_{i=1}^n (P_{c,i}) \subseteq \bigcup_{j=1}^m (P_{e,j}) \quad \text{avec} \quad m \geq n$$

Une telle contrainte fait de l'agence anthropique ce que l'on nomme un *anneau fermé*.

L'agent actuateur doit pouvoir signaler à l'agence anthropique le degré de difficulté d'actuation sur un paramètre. Pour satisfaire cette exigence, on assigne deux coûts à chaque paramètre. D'abord un coût appelé *coût intrinsèque* C_i , compris entre zéro et cent, qui représente la difficulté de modifier ce paramètre au sein du corps. Ensuite on assigne un coût appelé *coût extrinsèque* $C_e(i)$ ($0 \leq i \leq n$) pour un paramètre donné et pour un intervalle i donné de ce paramètre. Ce coût intrinsèque représente la difficulté de modifier un paramètre (pour des raisons physiques ou propres à l'agence) à un point spécifique de l'intervalle de variation d'un paramètre.

Enfin il faut encore prendre en compte la direction de modification d'un paramètre. Souvent certains paramètres ne peuvent être modifiés que dans

un sens dans l'organisme : on peut souvent augmenter la présence d'une substance sans être capable de la diminuer. Au coût intrinsèque et au coût extrinsèque, on ajoutera pour un paramètre générique e des coûts intrinsèques ainsi que extrinsèques vers le haut et vers le bas, i.e. $C_{i,down}$, $C_{e,down}(i)$, $C_{i,up}$ et $C_{e,up}(i)$.

Ces coûts intrinsèques et extrinsèques seront utilisés, comme nous allons le voir, pour l'évaluation des coûts de variation des agents extracteurs lors de la phase de coordination.

Pour un paramètre e et un nombre d'intervalles ni , l'ensemble des valeurs :

$$C_{i,down}, C_{e,down}, C_{e,down}(i), C_{i,up}(i)_{(0 \leq i \leq ni)}$$

forme le *modèle d'actuation* d'un agent actuateur pour ce paramètre e .

Spécification pour la coordination

Sans empiéter sur la description de la coordination (Section 2.4), nous voudrions néanmoins décrire ici les informations que l'agent actuateur rend disponibles aux agents décisionnels (au travers du KBB) pour mener à bien le mécanisme de coopération.

Tout d'abord, lorsqu'un agent décisionnel recherche un point candidat, il demande aux agents actuateurs concernés une estimation de la difficulté d'une variation positive ou négative d'un paramètre. Cette estimation permettra de décider quel point candidat, parmi ceux trouvés lors de la planification, sera effectivement proposé au KBB. Soit la position actuelle x d'un paramètre générique e , et soit up la valeur supérieure de l'intervalle total de variation de x et $down$ la valeur inférieure de l'intervalle total de variation de x . Cette estimation, dont la valeur varie entre zéro et cent, est calculée de la manière suivante :

$$C_{up} = \frac{\sum_{j=x}^{up} C_{e,up}(j) * p_j}{\sum_{j=x}^{up} p_j} * \frac{C_{i,up}}{100} \quad \text{avec} \quad e^{-\left(\frac{j^2}{2 * \left(\frac{up-x}{k}\right)^2}\right)}$$

$$C_{down} = \frac{\sum_{j=x}^{down} C_{e,down}(j) * p_j}{\sum_{j=x}^{down} p_j} * \frac{C_{i,down}}{100} \quad \text{avec} \quad e^{-\left(\frac{j^2}{2 * \left(\frac{down-x}{k}\right)^2}\right)}$$

Ces coûts C_{up} et C_{down} sont communiqués par les agents extracteurs lorsque les agents décisionnels cherchent un nouvel état.

Ensuite l'agent actuateur construit un vecteur de valeurs à chaque fois que le KBB ferme la session de coopération. A ce moment le KBB doit décider quel agent actuateur sera responsable de modifier la valeur d'un paramètre. Le KBB envoie à chaque agent actuateur une proposition d'actuation (i.e. la nouvelle valeur du paramètre) et chaque agent actuateur renvoie un *vecteur de coûts* qui permettra au KBB de choisir l'agent actuateur qui exécutera l'action. Ce vecteur de coût contient, pour chaque intervalle entre la valeur actuelle x_a et la valeur souhaitée x_s , une valeur entre zéro et cent indiquant la difficulté de mouvoir le paramètre au sein de l'intervalle dans la direction souhaitée. Chaque case représentant l'intervalle j du vecteur de coûts contient donc :

$$C_{direction}(j) = C_{e,direction}(j) * \frac{C_{i,direction}}{100}$$

Intégration de l'agent actuateur

L'agent actuateur s'intègre à l'agence anthropique en s'enregistrant aussi bien au KBB qu'au PBB. L'agent actuateur est comme l'agent décisionnel lié aux deux tableaux. La figure 2.1 du schéma logique de l'agence anthropique est donc incomplet puisqu'il ne met pas en évidence le fait que les agents actuateurs accèdent au PBB. Au KBB, l'agent actuateur signale sa présence, alors qu'au PBB, il s'inscrit pour la lecture des paramètres : l'agent actuateur a en effet besoin de savoir la valeur actuelle du paramètre qu'il contrôle, pour pouvoir construire, par exemple, le vecteur des coûts.

1. AA : accès au serveur du KBB
2. AA : demande d'un nouveau numéro d'identifiant au KBB
3. KBB : recherche d'un identifiant libre, univoque pour toute l'agence
4. KBB : renvoi de ce numéro d'identifiant
5. AA : réception et mémorisation de ce numéro
6. AA : envoi au KBB d'une référence à lui-même
7. KBB : mémorise l'association (identifiant, référence à cet AA).

Maintenant identifié au sein de l'agence, l'agent actuateur s'inscrit alors au PBB pour la lecture des ou du paramètre(s) qu'il contrôle.

1. AA : demande d'un paramètre au PBB
2. PBB : renvoi de l'identifiant de ce paramètre et inscription de l'agent dans la table des inscrits pour ce paramètre

3. AA : mémorisation de l'association (paramètre, identifiant du paramètre)
4. AA : communique ses capacités d'actuation au PBB sur le ou le(s) paramètre(s) qu'il contrôle
5. PBB : mémorisation des capacités d'actuation de cet agent sur les paramètres.

Notons que les informations données au PBB sont aussi accessibles au KBB.

En ce qui concerne l'interaction avec le PBB, l'agent actuateur y lit la valeur des paramètres qu'il contrôle. Au moment de la phase de coopération, il les lit sans aucune fréquence particulière, c'est-à-dire lorsqu'il en a besoin. En ce qui concerne l'interaction avec le KBB, l'agent actuateur fournit quelques services, comme nous l'avons vu, au moment où l'agence passe en phase de coordination. Le premier service est l'estimation des coûts pour les agents décisionnels, qui permet à ceux-ci de choisir le bon point candidat lors de la planification. Le second service est la construction du vecteur de coûts, qui permet au KBB d'assigner l'action sur un paramètre à l'agent actuateur le plus susceptible d'atteindre l'objectif.

Interaction avec le majordome

Le majordome, du point de vue de l'implémentation, interagit avec l'agent actuateur au travers du KBB.

L'agent actuateur offre divers services :

1. ajournement du modèle des coûts d'actuation (modification des intervalles et des coûts associés).
2. variation de la fréquence d'un paramètre au PBB en dehors de la phase de coordination
3. interruption ou activation dans une direction de la capacité d'actuation sur un paramètre
4. interruption ou activation dans les deux directions de la capacité d'actuation sur un paramètre.

2.3.4 Parameter BlackBoard

Dans cette section nous détaillerons la structure et le fonctionnement du PBB.

Rappelons que le rôle du PBB est de servir de noeud de communication entre les agents extracteurs et les autres agents de l'agence. Les agents extracteurs écrivent sur le PBB la valeur qu'ils viennent d'échantillonner, et cette valeur est à la disposition des agents décisionnels et des agents actuateurs. Il n'y a de cette manière aucun besoin de mécanisme de points d'attente entre les agents, qui peuvent agir indépendamment les uns des autres.

Le PBB contient une structure qui lui est propre contenant des couples (identifiant du paramètre, valeur actuelle).

Les éléments qui sont directement en lien avec le PBB sont les suivants :

1. les agents extracteurs
2. les agents décisionnels
3. les agents actuateurs
4. le majordome
5. la base de données des paramètres

Remarquons enfin que le PBB connaît la structure globale de l'agence à chaque moment, à savoir l'ensemble des agents qui sont connectés.

Insertion d'un paramètre

L'une des actions fondamentales dans l'agence est la mise à disposition pour les agents décisionnels et actuateurs des valeurs des paramètres. Les agents extracteurs, possédant après enregistrement l'identifiant de leur(s) paramètre(s), envoient régulièrement au PBB le couple (identifiant, valeur actuel du paramètre). Le PBB modifie la table des valeurs des paramètres : à l'entrée correspondant à l'identifiant reçu, le PBB efface l'ancienne valeur et la remplace par la valeur communiquée. Le PBB utilise aussi la nouvelle valeur pour la stocker dans la base de données.

Lecture d'un paramètre

La lecture d'un paramètre au sein du PBB est un mécanisme simple. Si un agent générique (décisionnel ou actuateur) souhaite lire la valeur d'un paramètre, il envoie une requête au PBB avec le numéro d'identifiant du paramètre. Le PBB, si l'identifiant existe, renvoie simplement le couple (identifiant du paramètre, dernière valeur du paramètre).

2.3.5 Knowledge BlackBoard

Le rôle du KBB, tout comme le PBB, est d'être un noeud de communication entre les agents décisionnels et actuateurs. Mais, contrairement au

PBB, sa fonction essentielle est de gérer la phase de coordination. C'est pour cette raison que nous le décrirons dans la section 2.4 relative au mécanisme de coordination.

2.3.6 Majordome

Le rôle du majordome est de servir de pont extérieur à l'agence anthropique et de permettre à l'utilisateur ou l'expert de superviser et de modifier l'agence anthropique. Il effectue ce rôle au travers du PBB et du KBB. Une série de service sont en fait offerts au majordome lorsqu'il se connecte au PBB. On peut considérer, d'un point de vue logique (voir figure 2.6), que le Majordome interroge l'agence au travers d'une interface qui elle-même opère sur une mémoire intermédiaire contenant l'ensemble des informations de l'agence.

Enfin nous résumons l'ensemble des fonctions que nous avons répertoriées pour les différents agents et les informations accessibles pour chaque paramètre dans le figure 2.5.

Agent extracteur	<ol style="list-style-type: none"> 1. ajout ou élimination d'un algorithme d'extraction 2. ajournement d'un algorithme d'extraction 3. interruption de l'activité d'extraction d'un paramètre 4. variation de la fréquence d'extraction d'un paramètre 5. variation de la fréquence d'échantillonnage d'un signal.
Agent décisionnel	<ol style="list-style-type: none"> 1. choix du seuil de bien-être 2. choix du seuil d'erreur pourcentuelle 3. ajournement du modèle de l'agent décisionnel 4. interruption de l'activité de planification 5. variation de la fréquence de prélèvement d'un paramètre
Agent actuateur	<ol style="list-style-type: none"> 1. ajournement du modèle des coûts d'actuation 2. variation de la fréquence d'un paramètre au PBB en dehors de la phase de coopération 3. interruption ou activation dans une direction de la capacité d'actuation sur un paramètre 4. interruption ou activation dans les deux directions de la capacité d'actuation sur un paramètre
Paramètre	<ol style="list-style-type: none"> 1. identifiant 2. nom 3. identifiant de l'agent extracteur qui l'extraie 4. unité de mesure 5. période d'extraction 6. numéro d'inscrits pour ce paramètre 7. liste des agents inscrits 8. fréquence d'extraction 9. graphique temporel ...

FIG. 2.5 – Liste des fonctions proposées par le majordome.

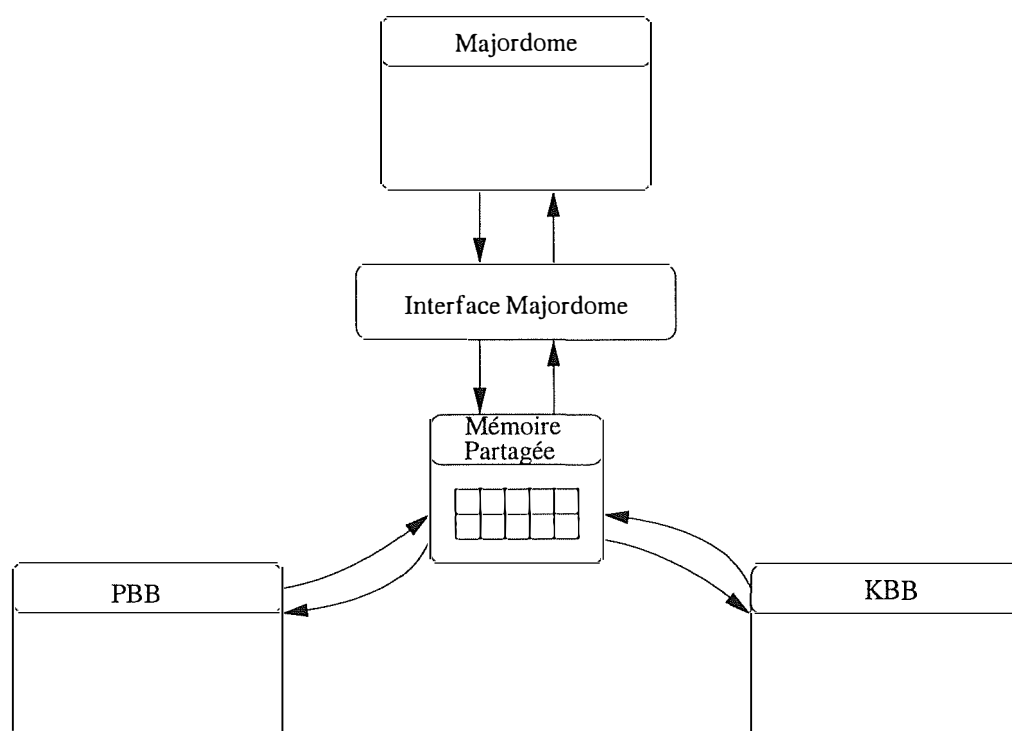


FIG. 2.6 – Schéma logique de l'interaction du majordome avec l'agence.

2.4 Le mécanisme de coordination

Décrivons maintenant le mécanisme fondamental de coordination. C'est le coeur de l'agence anthropique qui nous donnera un exemple concret de l'interaction entre les agents et de la possibilité de multi-modélisation au sein de l'agence.

2.4.1 Spécification du problème

Formulons, avant de lui donner une solution, le problème de la coordination des agents au sein de l'agence anthropique. Il faut distinguer trois termes qui, au sein de l'intelligence artificielle distribuée (DAI), ont chacun leur signification propre. La *coordination* entre des entités est un ensemble de règles permettant à un ensemble d'entités communiquant entre elles d'atteindre un objectif. Des agents dits *antagonistes* sont des agents ayant des objectifs contradictoires. La *coopération* est une coordination entre des agents non antagonistes. La *compétition* est une coordination entre des agents antagonistes. Ces définitions sont illustrées dans le figure 2.7.

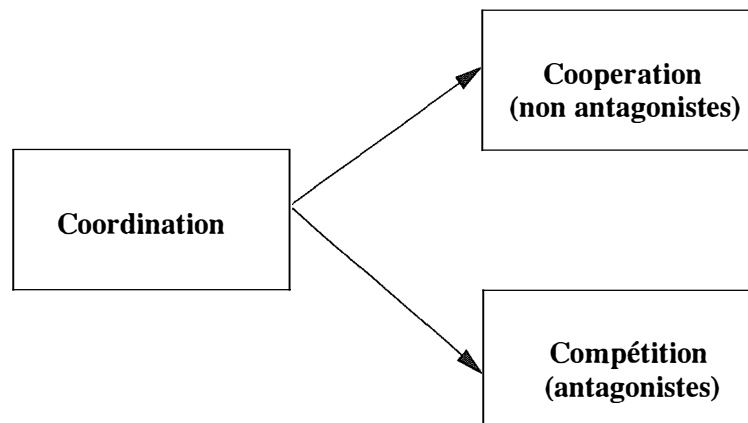


FIG. 2.7 – Coordination, coopération, et compétition.

Le problème à résoudre est de trouver un mécanisme de coordination entre les agents décisionnels et les agents actuateurs qui leur permettent d'atteindre un accord effectivement réalisable par les actuateurs. L'agence anthropique, sans ce mécanisme de coordination, est une agence incomplète représentée dans la figure 2.8.

La difficulté principale de ce problème est le manque de formalisation de l'agence anthropique. Cette difficulté provient du caractère expérimental à la fois de l'agence et de l'agence anthropique.

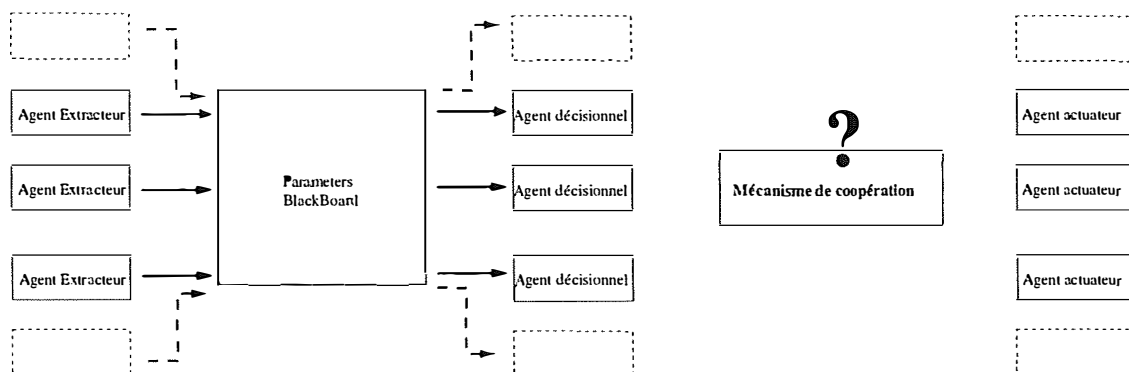


FIG. 2.8 – L’agence anthropique sans mécanisme de coordination.

2.4.2 Idée de la solution

Nous avons précisé le problème global ; nous allons maintenant essayer de présenter la solution retenue au sein de l’agence.

Le mécanisme de coordination va dépendre directement de la distribution de la connaissance entre les agents, plus précisément de la connaissance du modèle d’actuation des agents actuateurs (voir section 2.3.3). On peut en distinguer deux :

1. Dans le cas où les agents décisionnels connaissent le modèle des actions des agents actuateurs, les agents décisionnels effectuent une exploration de groupe : chaque agent décisionnel utilisera les modèles d’action dont il dispose pour atteindre un état meilleur pour l’agence entière. Des états, jugés optimaux par certains agents décisionnels, sont donc publiés sur un tableau intermédiaire de sorte que les agents décisionnels puissent évaluer ces états ou chercher à partir de ceux-là. Le travail des agents décisionnels est donc double : rechercher des états optimaux et évaluer les points proposés par les autres agents sur le tableau. Lorsqu’au moins un état satisfait tout le monde ou après un certain temps, le tableau choisit un état et demande aux agents actuateurs de l’atteindre.
2. Dans le cas où les agents décisionnels ne connaissent pas le modèle des actions des agents actuateurs, ils travaillent uniquement sur base de leur propre désir. Ils disposent d’un modèle partiel complètement disponible, dans lequel la recherche n’est pas liée aux actions possibles. Les désirs des agents décisionnels (i.e. les états optimaux qu’ils ont trouvés) seront alors l’entrée d’un mécanisme capable de trouver un état objectif qui maximise le bien-être de l’agence sur base des désirs des agents décisionnels. Une fois ce stade objectif trouvé, il reste à faire

exécuter cet état objectif aux agents actuateurs, c'est-à-dire d'attribuer les actions nécessaires aux différents agents.

Le mécanisme choisi sera le second, car le premier mécanisme proposé impliquerait une trop grande complexité. En effet il faut maintenir l'exigence de modularité de l'agence, et il faudrait dans la première solution construire des nécessaires fonctions d'ajournement des modèles d'actuation compliquant le projet ultérieur. Pour garder une modularité nette, il est préférable de séparer autant que possible le fonctionnement des agents décisionnels et des agents actuateurs. Nous adopterons donc la seconde approche. Le tableau intermédiaire utilisé sera évidemment le KBB, comme nous l'avons nommé dans les sections précédentes.

Décrivons à présent cette seconde approche dans ses grandes étapes.

- **Déclenchement de la phase de coordination** : L'agence anthropique rentre dans la phase de coordination lorsqu'un des agents décisionnels transmet une demande d'actuation au KBB. Cette demande se produit lorsque l'évaluation de l'état de bien-être de cet agent est inférieur à son seuil (voir section 2.3.2). Le point est proposé aux autres agents décisionnels, qui, s'ils ne sont pas satisfaits par le nouveau point (leur bien-être est sous le seuil), votent également des points sur le KBB.
- **Phase de l'objectif commun** : Le KBB prend une décision, sur base des propositions des agents, maximisant le bien-être total des agents décisionnels. Cette partie du KBB gérant la prise de décision globale est appelée *equalizer*.
- **Phase d'attribution des actions** : Lorsque l'*equalizer* choisit un état objectif à atteindre et ferme alors la coordination entre les agents décisionnels, le KBB devient alors responsable d'attribuer ces actions aux différents agents actuateurs, de manière la plus efficace possible. Cette partie du KBB est appelée *problem solver*.
- **Fin de la phase de coordination** : Lorsque le *problem solver* a décidé quelles actions étaient attribuées à quels agents actuateurs, il demande aux agents actuateurs d'exécuter ces ordres. La phase de coordination est terminée.

Soulignons que la phase de coordination se déroule en deux temps : une phase de coordination proprement dite, qui vise à trouver un accord commun entre les agents décisionnels, et une phase d'attribution des actions. Ces deux phases sont indépendantes, dans le sens précis que l'output de la première phase est l'input de la seconde. La figure 2.9 présente l'agence anthropique

complétée du mécanisme de coordination que nous avons choisi.

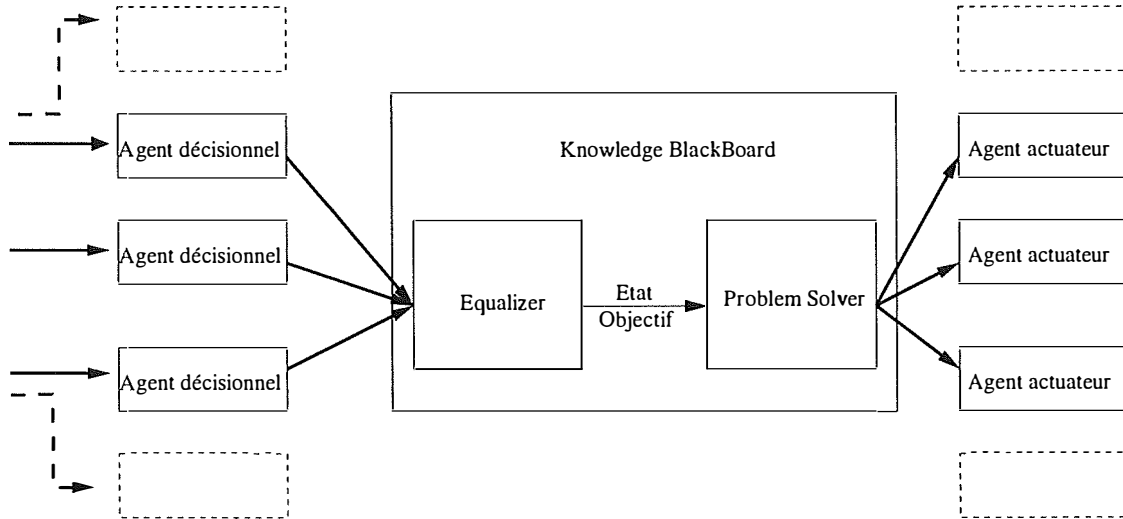


FIG. 2.9 – L’agence anthropique avec le nouveau mécanisme de coordination.

2.4.3 Objectif commun

Dans cette section nous décrirons en détail le fonctionnement de l’équalizer visant à aboutir à un accord commun entre les agents. Il s’agit de la première partie de la phase de coordination.

Choix de l’état objectif

L’état objectif, recherché par l’équalisateur, est l’état devant maximiser la satisfaction des agents décisionnels. Comment cet état est-il calculé? Supposons que les agents décisionnels votent chacun une liste d’états et que les agents extracteurs fournissent n paramètres (p_1, \dots, p_n) . Pour un agent décisionnel j ayant un modèle à m dimensions, un état est un point (v_{1j}, \dots, v_{mj}) . Chaque agent décisionnel vote alors régulièrement des points $V = (v_1, \dots, v_m)$ associé à un bien-être b compris entre zéro et cent. L’équalizer possède une liste $[(V_1, b_1), (V_2, b_2), \dots]$ où (V_j, b_j) est l’état souhaité par l’agent décisionnel j associé avec le bien-être b_j de cet état.

L’équalisateur va effectuer une *moyenne pondérée* pour chaque paramètre qu’un agent décisionnel a souhaité modifier. Si na est le nombre d’agents, alors la valeur du paramètre P_i se calcule comme suit :

$$P_i = \frac{\sum_{j=1}^{na} v_{ij} b_j}{\sum_{j=1}^{na} b_j}$$

La figure 2.10 exprime graphiquement le processus de choix de la valeur d'un paramètre.

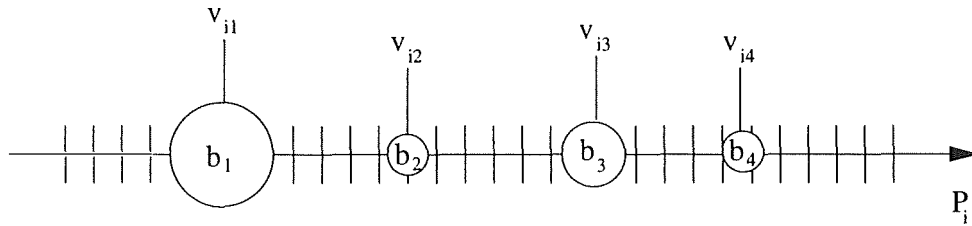


FIG. 2.10 – Représentation graphique du calcul de la valeur de P_i .

Modification du schéma

Le mécanisme de coopération tel que nous l'avons exposé jusqu'ici pose un problème majeur : les agents décisionnels, lorsqu'ils ont publié un état souhaité sur le KBB, ne peuvent pas rechercher d'alternatives meilleures. Un autre problème, lié à la décision de ne pas donner la connaissance des modèles d'actuation, réside dans le fait que les agents décisionnels font une recherche dans des directions qui ne sont pas accessibles par les agents actuateurs. Par exemple, les agents actuateurs peuvent être capables uniquement d'augmenter un paramètre. Dans ce cas, l'agent décisionnel peut se passer d'explorer tout un semi-espace de son modèle partiel.

Pour outrepasser ces limitations, le flux de messages ne va plus seulement des agents décisionnels au KBB, mais on introduit également un flux de messages du KBB vers les agents décisionnels. D'abord, l'équalizer communique l'état actuel qu'il a calculé pour permettre aux agents décisionnels de rechercher d'autres points et de faire de meilleures propositions. Ensuite, l'équalizer donne régulièrement l'objectif commun actuel au problème à résoudre de telle sorte que celui-ci puisse renseigner sur la difficulté d'actuation des paramètres ; cette difficulté d'actuation est liée aux coûts C_{up} et C_{down} élaborés par les agents actuateurs, déjà présentés dans la section 2.3.3. De plus, les agents décisionnels connaissent les directions d'actuation des paramètres.

Ces nouveaux flux de messages sont représentés dans la figure 2.11.

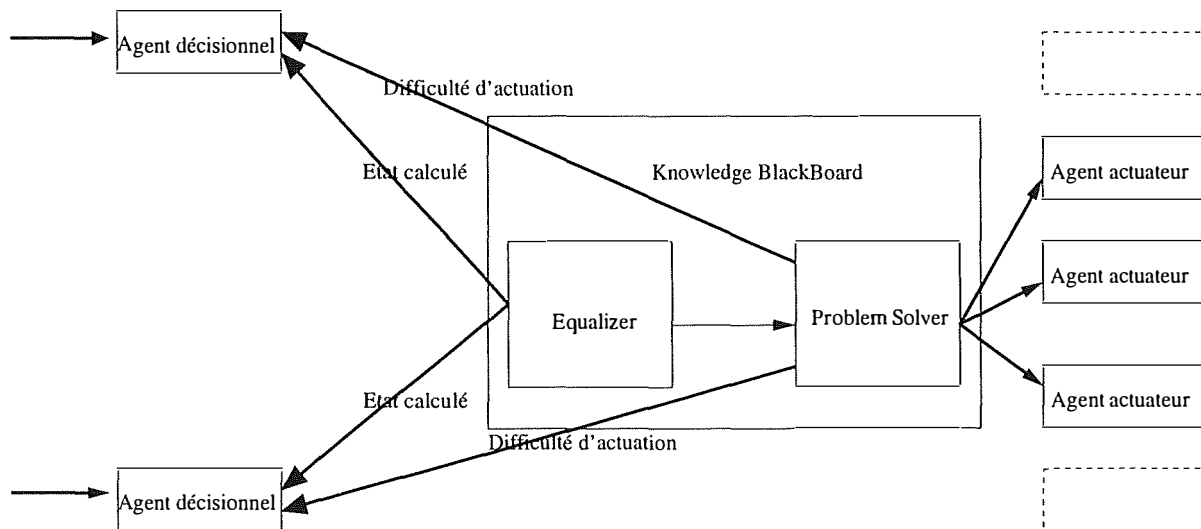


FIG. 2.11 – Nouveaux flux dans le mécanisme de coordination.

Assistant de l'agent décisionnel

Nous avons évité de détailler, dans la section 2.3.2 sur les agents décisionnels, le rôle de l'*assistant* au sein de l'agent décisionnel (voir figure 2.3). Son rôle est de choisir parmi l'ensemble des points trouvés par le planificateur le point ayant le coût le plus bas possible ; c'est en effet un tel point qui aura le plus d'influence sur la décision de l'équalizer.

Comment l'assistant évalue-t-il le coût d'un état nouvellement trouvé ?

$$\text{État actuel} \quad S = (x_{1S}, \dots, x_{NS})$$

$$\text{État souhaité} \quad T = (x_{1T}, \dots, x_{MT})$$

$$\text{Coût de la variation} \quad C\Delta_j = CV_j \quad |x_{jT} - x_{jS}|$$

avec CV_j coût de la variation infinitésimale

$$\text{Coût du point} \quad C_T = C\Delta x_1 + \dots + C\Delta x_M$$

Ainsi l'introduction du coût pour un point permet d'éviter la recherche anarchique au sein du modèle interne d'un agent décisionnel et d'augmenter les chances des agents décisionnels d'être satisfaits, et donc d'obtenir un bien-être total meilleur pour l'agence.

Il reste à définir CV_j . Trois types de coûts sont pris en compte pour un paramètre donné :

1. *coût institutionnel* : coût qui représente la difficulté ou la facilité avec lequel un paramètre peut être modifié en général. Le coût institutionnel est fixé par un expert.
2. *coût de direction* : coût qui représente la difficulté instantanée rencontrée par le problem solver; celui-ci fixe ce coût lorsqu'il interagit avec les agents actuateurs. Il s'agit des coûts C_{up} et C_{down} . Par exemple, le fait d'abuser d'un actuateur sur un paramètre augmente le coût de direction lié à ce paramètre.
3. *coût de coordination* : représente une mesure de la difficulté à faire varier le résultat à cause des désirs des autres agents. Ce coût dépend donc de la somme totale des poids associés aux désirs exprimés sur le paramètre. Si l'on se réfère à la figure 2.10, on voit que ce coût dépend de $b_1 + b_2 + \dots$.

La figure 2.12 résume les flux existant entre les différentes entités de l'agence anthropique.

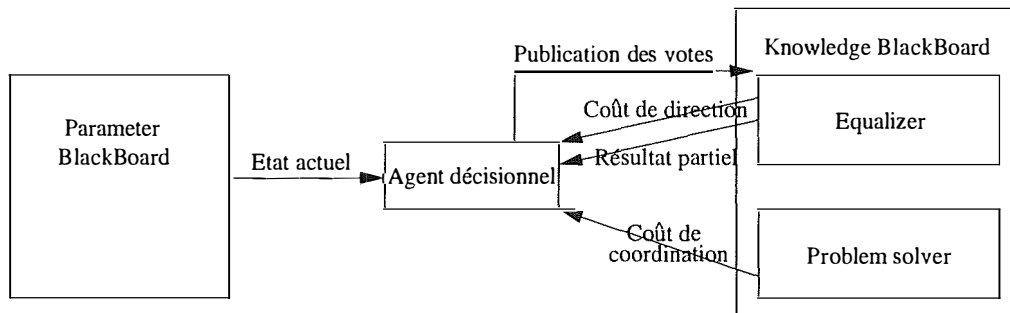


FIG. 2.12 – Résumé des flux entre les différentes entités.

2.4.4 Attribution des actions

Nous entrons maintenant dans la seconde phase du mécanisme de coopération. L'équalizer a décidé d'un état objectif, et celui-ci est passé au problem solver. Le problème du problem solver est maintenant de répartir les actions aux différents agents actuateurs de manière efficace.

Uniformisation des intervalles

D'abord le problem solver considère chaque paramètre, et demande aux agents actuateurs concernés par ce paramètre de lui envoyer leur *vecteur de*

coûts (voir section 2.3.3). Comme la découpe en intervalles diffère d'un agent actuateur à l'autre, il faut uniformiser les intervalles de telle sorte qu'une comparaison soit possible. Pour ce faire, le problem solver découpe chaque vecteur de coûts en croisant les intervalles (voir figure 2.13).

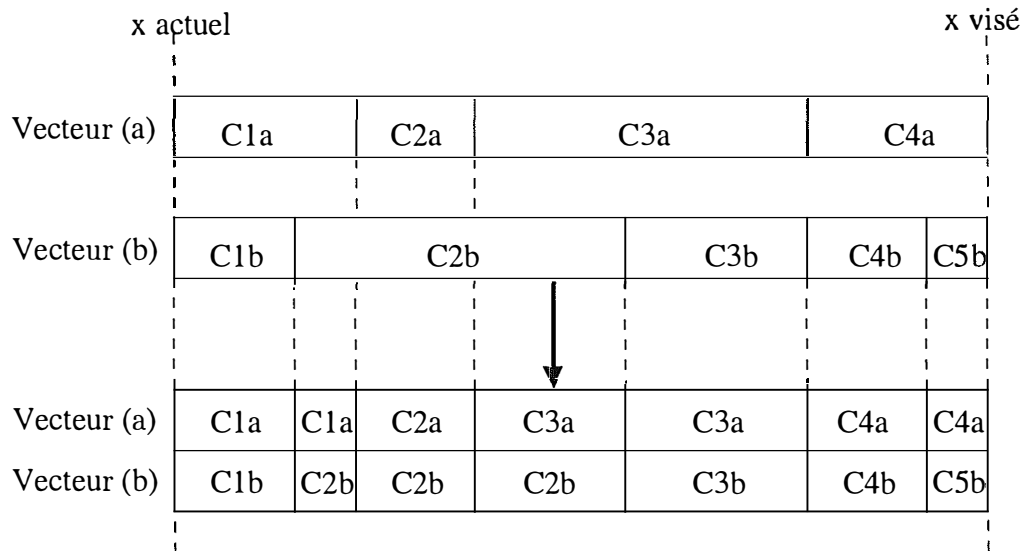


FIG. 2.13 – Schéma de la découpe des intervalles par le problem solver pour un paramètre donné.

Assignment des actions

Le problem solver doit maintenant assigner les actions proprement dites.

Deux exigences doivent être remplies pour l'assignation des actions. D'abord ces actions doivent être assignées de manière efficace, impliquant le moins d'effort de la part des actuateurs matériels, c'est-à-dire que les actions doivent être assignées aux coûts les plus bas possibles. Ensuite il faut que l'algorithme d'assignation des actions rende une solution univoque.

Supposons, étant donné l'état actuel x_a , que l'équalizer aie fixé un état objectif x_r . Les objectifs d'efficacité et d'unicité sont alors atteints grâce à ces trois critères appliqués à chaque paramètre :

- *Critère du coût minimal* : le problem solver trouve un vecteur de coûts où chaque intervalle a un coût inférieur aux autres vecteurs ; l'agent actuateur relatif à ce vecteur de coût est alors choisi (voir figure 2.14(a)).
- *Critère de cohérence* : si le premier critère échoue à désigner un agent actuateur, l'action est assignée à l'agent actuateur qui a répondu avec

un coût inférieur sur un nombre supérieur d'intervalles (voir figure 2.14(b)).

- *Critère d'exécution partielle* : si le dernier critère échoue, l'action est assignée seulement à l'agent qui a répondu avec un coût minimal sur la première partie des intervalles. Si deux ou plus agents actuateurs sont candidats à ce critère, on en choisit un au hasard, par exemple celui ayant le numéro d'identifiant le plus bas (voir figure 2.14(c)).

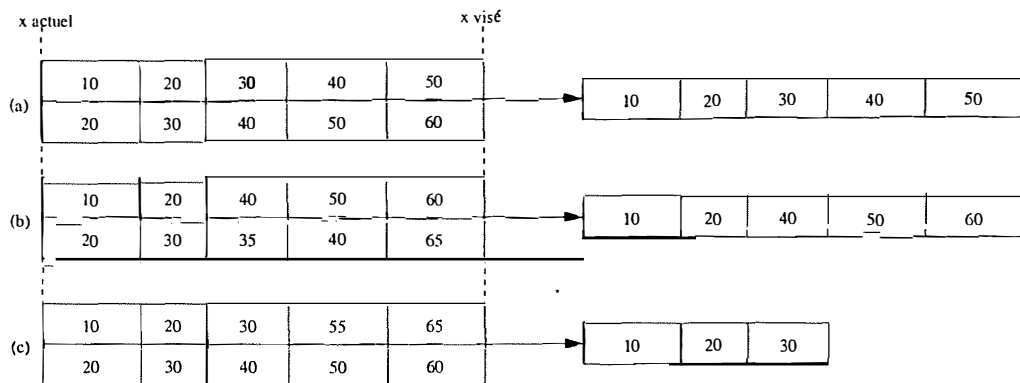


FIG. 2.14 – Critères d'assignation des actions : (a) coût minimal, (b) cohérence, (c) exécution partielle.

Il apparaît que, sur base de ces principes, un seul agent sera toujours choisi. Il se peut que le choix ne soit pas le plus efficace comme il apparaît si l'on considère un vecteur de coût $[1, 1, 1000, 1000, 1]$: cet agent actuateur gagnerait sûrement grâce au second critère, mais la somme totale des coûts est supérieure à celle d'un autre agent actuateur (par exemple $[30, 30, 30, 30, 30]$). Néanmoins ces cas sont extrêmement rares et peuvent être négligés, la difficulté d'actuation étant généralement progressive d'un intervalle à l'autre. Un vecteur de coûts de type $[1, 1, 1000, 1000, 1]$ est totalement improbable.

Remarquons également que, si le troisième critère est atteint, l'action peut n'être exécutée que partiellement, sur le ou les premiers intervalles d'un agent actuateur. Dans ce cas, l'action sera poursuivie éventuellement lors de la prochaine phase de coordination.

Les actions sont maintenant réparties, et le problème solver demande aux agents actuateurs d'effectuer ces actions.

Chapitre 3

Générateur de représentation de modèles physiologiques pour une approche distribuée

3.1 Introduction

Dans ce chapitre nous allons décrire le système de génération de modèles pour les agents décisionnels de l'agence anthropique. Ce système a été développé par l'auteur ainsi que Nicola Gatti [13] pour permettre la génération aisée de nouveaux modèles, que l'on puisse introduire dynamiquement dans l'agence anthropique.

3.2 Spécification du problème

Rappelons tout d'abord que chaque agent décisionnel dans l'agence possède un modèle qui est une représentation d'un phénomène physiologique du corps humain. En particulier, nous avons utilisé un modèle qui surveille le taux de glucose et d'insuline dans le corps. L'agent décisionnel lit sur le PBB de l'agence les paramètres pertinents, en l'occurrence le taux de glucose et le taux d'insuline, et vérifie au moyen du modèle si ces taux sont admissibles, à savoir si ces taux correspondent bien à un état jugé non pathologique. La section 2.3.2 (p.37) du chapitre 2 a déjà abordé succinctement le problème de la représentation du modèle, nous allons à présent détailler la forme et l'utilisation de ce modèle.

Le problème général peut être formulé comme la modélisation d'un système de représentation d'un modèle physiologique dans le cadre nouveau des systèmes distribués, plutôt que dans le cadre classique des systèmes de

contrôle monolithiques (voir le chapitre 1 et le développement sur le concept d'Agence).

Le problème consiste donc à fournir une représentation d'un modèle d'un comportement physiologique du corps humain, de telle sorte que l'agent décisionnel, pour un état donné, c'est-à-dire pour des valeurs données à un ensemble de paramètres qui sont l'entrée du modèle, sache si cet état est sain ou pathologique. Une fois obtenue une représentation capable de fournir cette fonctionnalité, nous aurons atteint notre objectif.

Avant de tenter de décrire une solution, il faut d'abord connaître les caractéristiques d'un modèle physiologique.

3.3 Modèle physiologique

Les modèles physiologiques mathématiques possèdent plusieurs particularités importantes à souligner pour élaborer une solution à notre problème de départ.

1. L'entrée des modèles physiologiques sont des courbes par rapport au temps. Par exemple, on aura une courbe du taux de nourriture en fonction du temps en entrée.
2. La sortie des modèles physiologiques sont également des courbes, qui donnent par exemple le taux de glucose par rapport au temps.
3. Le modèle physiologique dépend également, outre du temps, de paramètres physiologiques. Ces paramètres des modèles représentent les particularités propres à chaque corps. En particulier, les paramètres de modèle initiaux du modèle physiologique sont considérés comme sains, et plus on s'en éloigne, plus le modèle produit des courbes pathologiques.

Le modèle physiologique est par conséquent *dynamique*, c'est-à-dire que son output ne dépend pas seulement de l'entrée, mais également de paramètres de modèle externes.

A ce stade, on peut entrevoir une solution simple : lorsqu'un agent lit les paramètres qu'il surveille, il peut, pour savoir si ces paramètres se trouvent dans l'ensemble des solutions du modèle, calculer toutes les solutions et vérifier si l'état s'y trouve. Un tel calcul nécessite de calculer la sortie du modèle physiologique pour chaque input et pour chaque ensemble de paramètres de modèle possibles. Cette solution n'est donc pas souhaitable, car elle demanderait de la part des agents décisionnels des temps de calcul beaucoup trop élevés.

Nous devons, pour éviter cette difficulté, produire une représentation statique d'un modèle dynamique, de telle sorte qu'en un et un seul accès un agent puisse connaître le degré de pathologie de l'état considéré.

Il faudrait donc générer préalablement l'ensemble des solutions du modèle, pour chaque courbe d'entrée et pour chaque ensemble de paramètres. Les paramètres ont des valeurs finies, et il est assez simple de générer l'ensemble des valeurs de ces paramètres de modèle.

Mais qu'en est-il des courbes d'entrée? Pouvons-nous les connaître toutes? Oui. Une étude approfondie des courbes d'entrée montre qu'elles ne diffèrent que par l'amplitude [13, p.5]. La figure 3.1 en donne une illustration. On peut dès lors générer l'ensemble des courbes de sorties du modèle physiologique, et s'en servir pour former un modèle statique à usage d'un agent décisionnel.

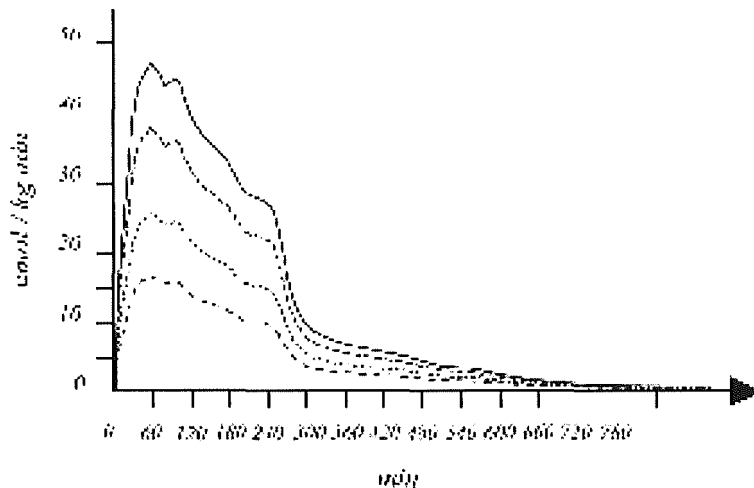


FIG. 3.1 – Les différentes courbes d'input ne diffèrent que par l'amplitude.

Si nous pouvons déterminer les solutions du modèle physiologique, il reste à lier ces solutions avec le taux de pathologie dont l'agent décisionnel a besoin. Or les paramètres de modèle nous donnent cette indication. On connaît l'ensemble des paramètres sains, auquel on fera correspondre une valeur entière nulle. On sait également que plus les paramètres physiologiques de modèle diffèrent de la valeur des paramètres de modèle sains, plus les solutions correspondent à des états pathologiques. On peut donc lier ces solutions à des valeurs entières supérieures, et ce de manière progressive à mesure que les paramètres de modèle s'éloignent des paramètres de modèle sains.

Après analyse de la structure générale d'un modèle physiologique, nous possédons une solution théorique pour la représenter au sein d'un agent décisionnel. Il reste à construire effectivement cette représentation, et ce sera l'objet de la section suivante.

3.4 Génération de la représentation du modèle

La génération de la représentation du modèle implique un problème d'implémentation, qui est la *discrétisation* des variables. La discrétisation est en effet nécessaire pour obtenir une représentation finie et utilisable.

Nous possédons, pour générer la représentation, d'un ensemble d'équations du modèle physiologique, et d'une relation entre le taux de pathologie et la variation des paramètres de modèle.

3.4.1 Espace multi-dimensionnel

La représentation statique adoptée sera un espace de solutions. Chaque dimension sera une variable de sortie du modèle physiologique. Pour un modèle physiologique donnant en sortie des courbes de concentration de glucose et des courbes de concentration d'insuline, nous aurons donc deux dimensions : le glucose et l'insuline.

Chaque dimension possède le nom de la variable, l'intervalle des valeurs de la variable, et les intervalles de discrétisation et la quantité de temps pour la discrétisation.

L'intervalle des valeurs est l'intervalle dans lequel les valeurs de cette variable peuvent se trouver.

Les intervalles de discrétisation d'un variable sont les intervalles dans lesquels a été découpée la dimension de cette variable. Les intervalles de discrétisation sont rendus nécessaires par le fait qu'il est impossible de retranscrire la continuité des valeurs. De plus, certaines régions se révéleront vides d'information ; on pourra donc adopter un découpage plus fin dans les régions intéressantes, et un découpage plus grossier dans les autres régions.

La quantité de temps pour la discrétisation n'est nécessaire que pour les variables *calculées*. Une variable est dite calculée lorsqu'elle n'est pas mesurée directement par l'agence anthropique, mais lorsqu'elle est calculée à partir des variables mesurées. Par exemple, la variation de glucose peut être discrétisée toutes les trente secondes, ce qui implique que l'agent décisionnel ne devra recevoir cette variable calculée que toutes les trente secondes.

3.4.2 Génération des solutions

Nous possédons un ensemble d'équations pour le modèle physiologique et un ensemble de paramètres de modèles sains. Le comportement sain du modèle est obtenu à partir de ce modèle, et les comportements pathologiques sont obtenus en faisant varier les paramètres de modèle.

Soit x_i un vecteur d'entrée, représentant une des courbes d'entrée (voir figure 3.1). On les discrétise tout d'abord selon la discrétisation T choisie pour cette dimension. Les courbes d'entrée pourront par exemple être les courbes d'entrée de glucose de la figure 3.1, discrétisées selon les t_1, t_2, \dots, t_T . Soit P l'ensemble des paramètres de modèle. Pour *un et un seul* ensemble de valeurs des paramètres de modèle, la solution est une matrice S contenant pour chaque temps t_i la valeur des courbes pour ce temps :

$$S = \begin{pmatrix} y_1(t_1) & y_2(t_2) & \dots & y_1(t_T) \\ \dots & \dots & \dots & \dots \\ y_s(t_1) & y_s(t_2) & \dots & y_s(t_T) \end{pmatrix}$$

Les colonnes de la matrice S sont les vecteurs output de la première courbe discrétisée en temps t_1, \dots, t_T . Ces vecteurs output sont obtenus par calcul au travers du modèle mathématique dont les paramètres sont fixés par les paramètres de modèle.

Nous introduisons maintenant la notion d'*espace atteignable*, qui est un ensemble quelconque de points qui peuvent être solutions du modèle physiologique. Ainsi la matrice S est toujours un ensemble atteignable.

Le processus de calcul que nous venons d'exposer est valable pour un ensemble de paramètres; on réitère en fait ce processus en faisant varier les paramètres de modèle.

Tout le problème est de savoir comment donner une représentation de cet espace atteignable dans l'espace multi-dimensionnel.

3.4.3 Degré de pathologie

Nous avons besoin d'une relation entre le degré de pathologie et les paramètres de modèle. On va en fait faire correspondre à chaque matrice S un degré de pathologie qui dépendra des paramètres de modèle fixes associés à cette matrice S . Les paramètres de modèle initiaux sont les paramètres d'un corps sain; plus ces paramètres diffèrent des paramètres de modèle initiaux plus est élevé le degré de pathologie. Dès lors, pour une solution S donnée, nous lui associons un degré de pathologie correspondant aux paramètres de modèle utilisés pour générer cette solution S . Ce degré de pathologie est

également appelé *valeur potentielle*. Ainsi chaque espace atteignable possède une valeur potentielle qui lui est associée.

Notons également que par convention, un état sain est caractérisé par un degré de pathologie valant 0, tandis que l'état le plus pathologique vaut 100.

La fonction potentielle, assignant le degré de pathologie à la matrice S , dépend de pc_h les paramètres de modèle initiaux et de pc_p les paramètres de modèle pathologiques courants, où :

$$pc_h = (p_{h1}, \dots, p_{hZ})$$

$$pc_p = (p_{p1}, \dots, p_{pZ})$$

La fonction potentielle vaut alors :

$$f(pc_h, pc_p) = \alpha \cdot \prod_{i=1}^Z \frac{p_{hi}}{P_{pi}}$$

La fonction potentielle est un produit de rapport entre les valeurs des paramètres de modèle initiaux et les paramètre de modèle choisis, de telle sorte que la fonction potentielle soit la plus grande lorsque les paramètres choisis s'écartent des paramètres initiaux. Le facteur α est un facteur de normalisation qui permet de ramener le potentiel dans l'intervalle $[0,100]$.

3.4.4 Introduction des données

Le dernière étape est d'introduire ces solutions avec leur potentiel associé dans l'espace multi-dimensionnel.

Nous possédons deux vecteurs $(y_1(t_i), \dots, y_s(t_i))$ et $(x_1(t_i), \dots, x_s(t_i))$ ainsi que le degré de pathologie calculée par la fonction potentielle. Nous ne pouvons introduire simplement le degré de pathologie pour chaque couple de coordonnées $(y_j(t_i), x_j(t_i))$ car les dimensions sont discrétisées. Il faut prendre en compte cette discrétisation et faire correspondre ces valeurs aux intervalles.

La discrétisation des intervalles pour chaque dimension soulève trois problèmes principaux. Le premier problème est que l'on ne peut garantir que tout l'espace mutli-dimensionnel soit rempli. En effet, à cause de la discrétisation, certains points de l'espace multi-dimensionnel seront vides. Le second problème est la propriété de continuité du potentiel dans l'espace multi-dimensionnel. En effet, il est réaliste de penser qu'un bon degré de pathologie ne peut se trouver à côté d'un mauvais degré de pathologie. Le troisième problème est qu'un agent décisionnel recherche toujours le meilleur de degré de pathologie par une recherche descendante, suivant les points qui

ont le plus faible degré de pathologie. Introduire les degrés de pathologie tels quels impliquerait que les agents décisionnels seraient incapables d'effectuer une recherche ascendante.

Pour résoudre ces problèmes nous utilisons la notion de *déflagration*. Une déflagration pour un point dans l'espace multi-dimensionnel est un espace autour de ce point dans lequel la degré de pathologie augmente à mesure que l'on s'éloigne de ce point. Par conséquent on peut se représenter l'introduction des données dans l'espace multi-dimensionnel comme le bombardement d'une plaine. L'espace multi-dimensionnel vide peut être considéré comme une plaine, et chaque déflagration dans cet espace est comparable à l'explosion d'une bombe dans cette plaine.

Une déflagration dépend de trois facteurs : l'*intervalle de déflagration*, la *forme de déflagration* et le *profil de potentiel* dans la forme en fonction de l'intervalle.

Nous avons un algorithme dont la forme de déflagration est une sphère.

L'intervalle de déflagration est déterminé en fonction de l'«intensité» du degré de pathologie du potentiel. L'intervalle de déflagration est maximum lorsque le degré de pathologie du point est minimum (zéro) et minimum lorsque le degré de pathologie du point est maximum (cent). Autrement dit, l'intervalle de déflagration devrait être inversement proportionnel au degré de pathologie du point. Ce choix est justifié par le fait qu'il est réaliste de penser qu'autour d'un point sain il doit exister beaucoup de points dont le degré de pathologie est proche de zéro. Notons que l'intervalle total d'une déflagration dépend des dimensions de l'espace multi-dimensionnel : dans le cas contraire, il se pourrait qu'une déflagration recouvre l'ensemble d'une ou plusieurs dimensions.

Le profil de déflagration définit la courbe du «trou» créé dans l'espace. Il est clair que cette courbe de potentiel doit avoir une valeur minimum autour du centre de déflagration et une valeur maximale sur le bord de la déflagration. Nous avons considéré que cette courbe de potentielle était fonction de la distance entre le point et le centre de la déflagration.

Des conflits peuvent intervenir. Que se passe-t-il en effet si une déflagration recouvre d'autres points déjà placés dans l'espace multi-dimensionnel? Nous appelons un *point réel* un point qui provient des solutions générées par le modèle physiologique. Le critère que nous avons adopté pour résoudre ces conflits est de considérer que l'on n'écrasait pas les points réels, puisque ceux-ci proviennent directement du modèle physiologique et non d'une déflagration ; de plus, les points qui possèdent le plus haut potentiel sont conservés.

La génération de la représentation du modèle comprend trois étapes.

D'abord on construit un espace multi-dimensionnel discrétisé pour les variables d'entrée et de sortie. Ensuite pour chaque ensemble de paramètres fixés on calcule la sortie des courbes d'entrée, à laquelle est associée un degré de pathologie. Enfin, on introduit ce degré de pathologie aux coordonnées obtenues dans le modèle multi-dimensionnel, ce qui ne va pas sans problèmes lié à la discrétisation des intervalles, et qui nous a obligé à introduire le concept de déflagration. Les deux dernières étapes sont réitérées pour chaque valeur des paramètres de modèle possible.

3.5 Exemples de modèles

Nous donnons un exemple graphique du résultat du générateur à la figure 3.2. Le modèle représenté possède trois dimensions : glucose, insuline, et la différence de glucose avec le temps précédent. Chaque image représente un plan de découpe dans l'espace à trois dimensions.

3.6 Conclusion

Nous obtenons une représentation statique d'un modèle physiologique dynamique. Notre problème de départ était de fournir une représentation d'un modèle d'un comportement physiologie du corps humain de telle sorte que l'agent, sur base des paramètres lus, puisse savoir cet état est pathologique ou non. Notre objectif est atteint, puisque nous disposons d'un espace multi-dimensionnel dans lequel l'agent lit le degré de pathologie du point correspondant aux paramètres lus.

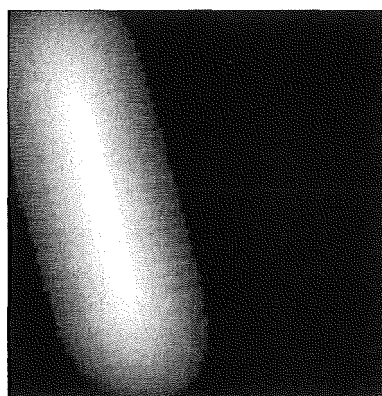
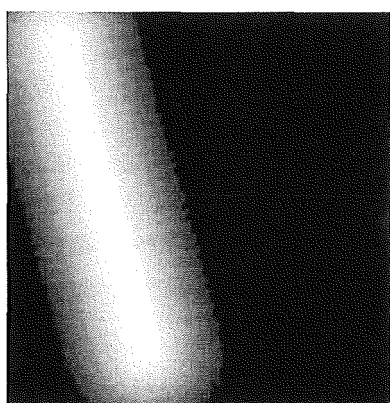
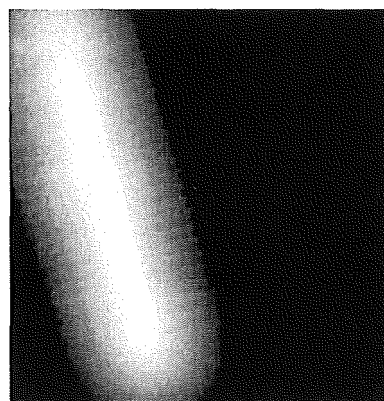
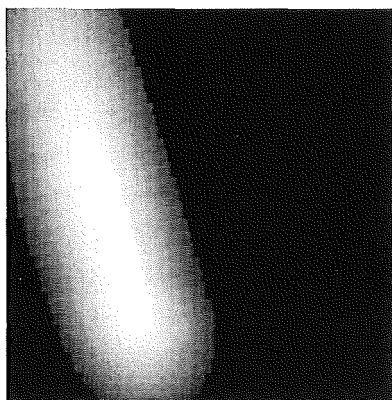
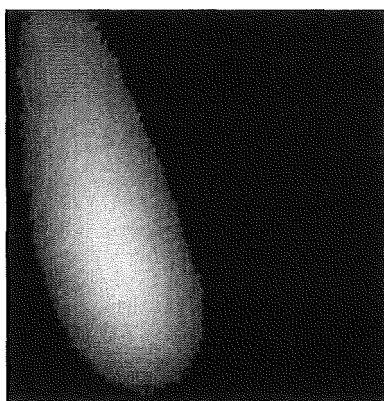
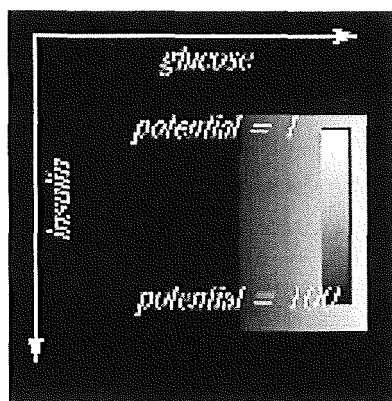


FIG. 3.2 – Découpe des plans de glucose-insuline à différentes valeurs de la variation de glucose.

Deuxième partie

Démarche épistémologique

Chapitre 4

Découverte scientifique

4.1 Introduction

Nous avons montré dans la première partie de ce travail les éléments techniques de l'agence, en développant notamment l'architecture concrète de l'agence anthropique. Nous entrons à présent dans la partie épistémologique en présentant les thèses épistémologiques des auteurs du concept d'agence. Ces thèses prennent racine dans le débat de la découverte scientifique. L'agence va en effet être vue comme une machine capable d'aider l'activité scientifique. Mais elle va surtout apporter une nouvelle utilité dans la découverte scientifique : elle servirait en effet de *description opérationnelle* du savoir scientifique. Elle aurait un double rôle, un rôle de soutien et un rôle de description, et ces deux rôles seraient, comme nous allons le voir, complémentaires. L'équipe de Milan va, de plus, développer une agence particulière, une agence représentationnelle, qui serait l'exemple même d'une machine intégrant ces deux rôles.

4.2 La découverte scientifique

4.2.1 Le débat

Une des différenciations entre IA faible et IA forte est la possibilité de faire réaliser la découverte scientifique à un ordinateur. D'un côté les partisans de l'IA faible considèrent cette activité comme complètement irrationnelle, et nient la possibilité de pouvoir la faire réaliser par une machine, d'un autre côté les partisans de l'IA forte considèrent cette activité comme totalement rationnelle, et affirment la possibilité de la faire réaliser par une machine. Le débat consiste donc en une possibilité d'une modélisation de l'activité de la

découverte scientifique, jusqu'ici seulement conduite par l'homme.

Bien sûr il nous faudrait donner une signification précise à la découverte scientifique, ce que nous ferons dans la section 4.2.2. Contentons-nous de préciser pour l'instant que la découverte scientifique permet de créer un nouveau savoir de la réalité.

Entre ces deux positions face au problème de la découverte scientifique, il est possible d'envisager une troisième voie, qui aborde sous un nouvel angle le rapport entre l'homme et la machine en assignant un nouveau rôle à la machine au sein de la découverte scientifique.

4.2.2 Cadre épistémologique de la découverte scientifique

L'équipe de Somalvico [11] propose de développer un cadre épistémologique en vue de mettre en évidence une alternative dans le débat à propos de la découverte scientifique. On retrouve le développement de ce cadre épistémologique dans bon nombre d'articles traitant de l'agence [9] [10] [8] [7]. Le but de la démarche est de préciser le rapport entre homme et machine et par là de départager le rôle de l'homme et de la machine dans la découverte scientifique. Il s'agit également de comprendre la découverte scientifique. Quel rôle attribuer à la machine dans la découverte scientifique ? Qu'est-ce que la découverte scientifique ? Le développement suit le plan suivant :

- **Une étude de l'activité scientifique**, pour comprendre l'acte de modélisation, pour mettre en place des concepts relatifs à la connaissance, et par là proposer des mécanismes qui seraient essentiels à la découverte scientifique. Cette étude se basera sur la façon dont Galilée comprenait la connaissance scientifique. Les concepts évoqués dans ce schéma sont fondamentaux et seront utilisés continuellement durant la progression vers le nouveau rôle des machines dans la découverte scientifique.
- **Une caractérisation de l'intelligence**, en exprimant l'existence d'une intelligence fabricatrice d'une part et créatrice d'autre part. C'est un premier pas vers une différenciation de l'intelligence humaine et de l'intelligence de la machine, et vers une définition de la découverte des machines.
- **Le lien entre machine et intelligence humaine**, où la machine est vue comme une réification d'un modèle.

- Une définition de la découverte des machines et de la découverte scientifique.
- La place de l'homme et de la machine, où l'homme est vu comme constitué de deux pôles, qui mettent en évidence la place de la machine dans son rapport avec l'homme, et qui permettra alors de d'identifier le rôle des machines dans la découverte scientifique.
- Un nouveau rôle dédié aux machines dans la découverte scientifique, conséquence des considérations sur le lien entre la machine et l'homme, rôle qui sera en fait double et qui dépasse l'opposition entre ceux qui affirment l'impossibilité pour une machine de mener la découverte scientifique et ceux qui affirment l'inverse.

Paradigme de Galilée

L'équipe de Somalvico propose ainsi d'utiliser le schéma général de la connaissance de Galilée [12]. Ce schéma (voir la figure 4.1) peut être considéré comme une représentation de la méthode scientifique.

Une première différenciation s'effectue entre la réalité et la connaissance de cette réalité. La réalité est vue comme un ensemble de phénomènes que l'homme peut percevoir. L'homme prend comme point de départ un phénomène dans la réalité, en conçoit un modèle, qui se retrouve dans la connaissance de la réalité. Ce passage de la réalité à la connaissance de la réalité s'appelle *abduction*. Il se forme alors un écart entre deux entités différentes : réalité et connaissance de cette réalité.

Lorsque nous nous plaçons du point de vue de la connaissance de la réalité, l'homme peut manipuler sa connaissance de la réalité *à priori*, c'est-à-dire indépendamment de l'expérience de cette réalité. L'homme peut alors produire une loi, qui est un nouvel élément de connaissance de la réalité. L'*induction* permet à l'homme de créer une loi plus abstraite que le modèle et la *déduction* lui permet de créer une loi moins abstraite que le modèle. L'ensemble des deux activités, induction et déduction, se nomme *inférence*.

Lorsque l'homme possède un nouvel élément de connaissance à partir d'un modèle, il peut tenter de confronter ce nouveau savoir avec la réalité, principalement pour prévoir un nouveau phénomène dans la réalité. Cette confrontation est appelée *adduction*, parce qu'elle conduit la loi vers le phénomène. L'adduction et l'abduction se nomment *afférence*.

L'ensemble de ce processus de connaissance est lui-même sujet à une méta-activité appelée critique, qui consiste principalement à améliorer le cycle abduction-inférence-adduction par rapport au cycle précédent, en vue d'obtenir une abduction plus efficace et une adduction plus satisfaisante.

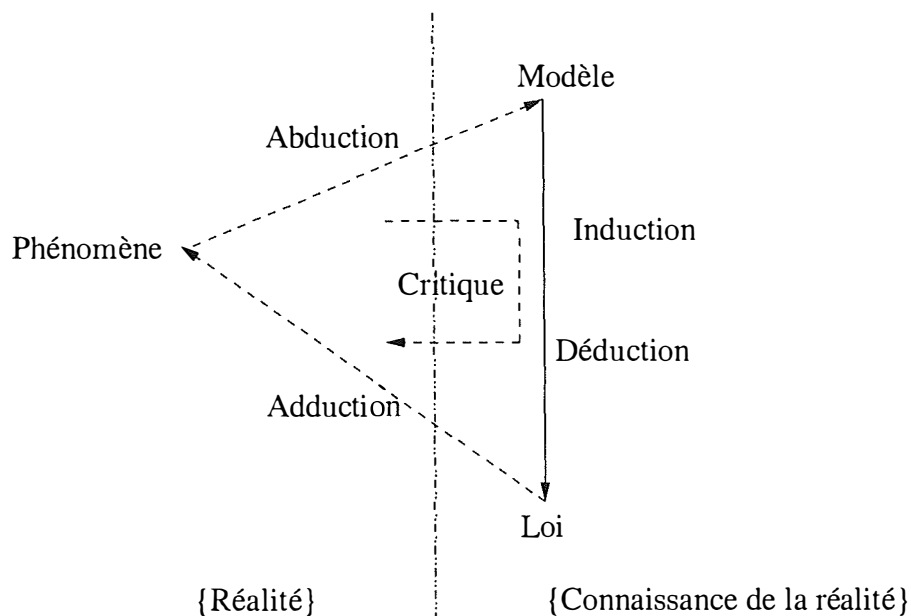


FIG. 4.1 – Paradigme de la connaissance selon Galilée.

Deux types d'intelligence

L'équipe de Somalvico propose également d'utiliser une distinction établie par Bergson [5] en ce qui concerne l'intelligence. Selon lui, l'intelligence peut être divisée en deux types : l'intelligence créatrice et l'intelligence fabricatrice. Lorsque l'homme observe la nature et qu'il invente un nouveau modèle, il exploite son intelligence créatrice, ce qui correspond au mouvement d'abduction du schéma de Galilée. L'intelligence créatrice est aussi en jeu lorsque l'homme prédit la venue d'un nouveau phénomène dans le monde. L'activité de critique, puisqu'elle concerne la modification de l'abduction et de l'adduction, appartient également à l'intelligence créatrice.

L'intelligence fabricatrice rejoint, quant à elle, l'activité d'inférence : l'homme induit ou déduit une loi d'un modèle.

Somalvico et son équipe précisent alors qu'ils considèrent que l'intelligence créatrice contient des éléments irrationnels qui ne peuvent pas être modélisés, tandis que l'intelligence fabricatrice contient des éléments rationnels qui peuvent être modélisés.

Machines et modèles

Quel lien existe-t-il entre une machine et son modèle ? La machine est une entité artificielle c'est-à-dire conçue et voulue par l'homme. Le point impor-

tant pour comprendre la notion de machine, poursuit l'équipe de Somalvico, est qu'une machine est toujours la *réification* d'un modèle. L'homme projette la loi qu'il a déduit d'un modèle dans une entité du monde qu'il a construit. La machine, alors, est définie comme un artefact réifiant un modèle. Entre la machine et le modèle il y a donc un rapport de concrétisation, c'est-à-dire que la machine rend réel le modèle seulement pensé.

Plus précisément, Somalvico remarque que d'une manière générale un modèle est composé d'une forme et d'un formalisme. La forme est la partie du phénomène que l'on choisit de modéliser, le formalisme est l'ensemble des moyens utilisés pour représenter la forme choisie. La forme représente donc l'abstraction effectuée sur le réel, c'est-à-dire le choix des éléments à retenir dans le phénomène. Autrement dit, la forme est le contenu de ce que l'on souhaite modéliser, le formalisme sont les moyens que l'on utilise pour modéliser ce contenu. L'acte de modélisation, de ce point de vue, peut être décrit comme l'acte de projeter la forme d'un phénomène dans un formalisme.

Comment alors concevoir qu'une machine puisse être la réification d'un modèle ?

Il faut noter que ces deux éléments, forme et formalisme, se retrouvent au sein de la machine elle-même. La forme, du point de vue du phénomène, consiste en la partie du phénomène que l'on souhaite modéliser dans la machine, la forme consiste donc en l'architecture de la machine, puisque les relations entre les composants de la machine déterminent l'objet que l'on représente. Le formalisme consiste quant à lui en les différents composants de la machine, puisque ceux-ci donnent l'ensemble des fonctions disponibles. Autrement dit, le formalisme de la machine est l'ensemble des moyens que nous possédons pour construire cette machine, et la forme de la machine est la manière dont on a agencé ses éléments (l'architecture) en vue de modéliser le contenu choisi du phénomène.

On comprend alors que la machine puisse être comprise comme une *réification* d'un modèle d'un phénomène.

Découverte des machines et découverte scientifique

Après ces considérations sur la nature de l'intelligence humaine, l'équipe de Somalvico peut alors préciser leur approche de la découverte scientifique. Si la machine est une réification des modèles créés par l'homme, et si l'homme ne peut pas modéliser son intelligence créatrice, la place des machines au sein de la découverte scientifique s'éclaircit : la découverte des machines concerne seulement la découverte que l'homme peut mener avec son intelligence fabricante.

La découverte scientifique peut ainsi être définie comme l'activité intel-

lectuelle conduite par l'homme lorsqu'il construit un modèle, en déduit ou induit une loi, et l'utilise pour prévoir l'apparition d'un nouveau phénomène (ce cycle peut être répété au moyen de la critique).

La découverte des machines, quant à elle, est définie comme l'émulation des activités intellectuelles fabricatrices, principalement le passage par l'induction et la déduction du modèle à la loi.

On voit que l'équipe de Somalvico a une position faible dans le débat de la découverte scientifique puisqu'ils affirment l'impossibilité de modéliser l'ensemble du processus intellectuel en jeu dans la découverte scientifique. Mais ils ne s'arrêtent pas là : ils vont préciser le rapport que la machine entretient avec l'homme et par là donner un rôle aux machines dans la découverte scientifique.

Homme bipolaire

En combinant le fait que l'intelligence créatrice existe seulement en l'homme et qu'une machine est une réification de la seule intelligence fabricatrice, Somalvico insiste sur l'émergence d'une nouvelle vision de l'homme. L'homme de ce point de vue peut être en effet décrit comme un bipôle compris comme des sites. Le premier pôle est le pôle homme-corps, qui représente le site naturel du corps de l'homme où les deux activités intellectuelles fabricatrices et créatrices sont utilisées. Le second pôle est le pôle homme-machine qui représente le site artificiel de la machine où seules les activités intellectuelles fabricatrices sont réifiées.

De plus, si nous suivons l'idée que seul l'homme possède l'intelligence créatrice, alors l'activité d'abduction, c'est-à-dire de créations de nouveaux modèles, appartient exclusivement à l'homme, et peut donc être réalisée seulement dans le pôle homme-corps. D'autre part, l'activité d'induction et de déduction peut être réifiée dans une machine et donc exécutée aussi bien par une machine que par l'homme.

L'homme bipolaire est donc l'homme considéré comme un double pôle, d'une part un pôle homme-corps où l'homme peut conduire tout type d'activité intellectuelle, et d'autre part le pôle homme-machine où l'homme peut faire exécuter, en la réifiant, son intelligence fabricatrice.

Une troisième voie dans le débat

Au-delà de la position de l'IA faible et de l'IA forte, une troisième voie se dessine. Sur base du paradigme de Galilée qui différencie afférence et inférence, et sur base de la différenciation entre intelligence fabricatrice et créatrice, l'équipe de Somalvico a montré que l'homme peut réifier une

partie de son intelligence en vue d'améliorer la découverte scientifique. L'homme est alors considéré comme bipolaire, c'est-à-dire comme un sujet composé de deux sites, l'un étant le corps naturel dans lequel l'homme peut utiliser son intelligence créatrice et l'autre étant le pôle artificiel dans lequel l'homme peut faire exécuter à la machine des activités de type fabricatrice.

L'équipe de Somalvico a ainsi déterminé ce qu'il faut entendre par découverte scientifique et découverte des machines, et a délimité le rôle exact que pouvaient avoir les machines dans la découverte scientifique : elles peuvent seulement mener des activités d'inférence. L'homme bipolaire exprime la dualité entre l'intelligence créatrice de l'homme et l'intelligence fabricatrice de la machine, et exprime la capacité pour l'homme d'extérioriser son intelligence fabricative dans la machine.

Quel peut être dès lors le rôle des machines dans la découverte scientifique ? On peut assigner à la machine, malgré qu'elle ne peut utiliser la créativité propre à l'homme dans la découverte, deux rôles : 1) soutenir la créativité dans la découverte scientifique et 2) décrire totalement le produit des actes créatifs impliqués dans la découverte scientifique.

4.2.3 Double rôle de la machine dans la découverte scientifique et leur interaction

Examinons, à présent, les deux rôles que la machine, comprise comme le site dans lequel l'homme fait accomplir des tâches de type fabricatrice, peut remplir dans le cadre de l'activité scientifique selon l'équipe de Somalvico, qui propose également une synergie entre ces deux rôles.

La machine comme soutien à la découverte scientifique

Le premier rôle que l'on peut assigner à la machine est celui de soutien dans la découverte scientifique. L'homme peut en effet être aidé et ainsi améliorer son activité scientifique au moyen d'une machine à information. L'homme peut adopter un pôle homme-machine puissant, pour exécuter des activités de type fabricatrice. Le site homme-machine est ainsi un pôle dans lequel l'homme délègue des activités fabricatrices et dont le rôle est de l'aider dans l'activité scientifique.

Sur base de cette relation entre l'homme et la machine l'équipe de Somalvico propose le concept de découverte scientifique bipolaire (DSB), qui envisage un scénario dans lequel la découverte scientifique est conduite par l'homme entendu comme un seul sujet, mais composé de deux pôles, qui sont des sites dans lesquels des activités sont effectuées.

La machine comme description du résultat de la découverte scientifique

Le second rôle de la machine est la description du résultat de la découverte scientifique. Cette proposition donne donc une description rationnelle du résultat de l'activité scientifique, plutôt que le processus lui-même. De cette manière, nous pouvons représenter les modèles obtenus comme résultat de la découverte scientifique sous la forme de machines. De plus, puisque cette description est basée sur la manière dont les machines fonctionnent, la description est purement opérationnelle, plutôt que dénotationnelle dans le cas d'un formalisme mathématique.

L'équipe de Somalvico propose comme machine de représentation l'agence elle-même pour investir le rôle de représentation de la machine, car elle possède des caractéristiques intéressantes permettant de mettre en lien des modèles d'un même phénomène. En effet une machine leur semble devoir posséder trois caractéristiques principales :

- cette machine devrait être capable de faire coopérer les différents modèles, c'est-à-dire des entités qui incluent les modèles issus de la découverte scientifique.
- les modèles coopérant de cette machine ne devraient pas être élémentaires, puisque les modèles scientifiques sont évidemment des représentations non élémentaires.
- cette machine devrait avoir un but unique, c'est-à-dire un but défini avant la construction de la machine, parce que le second rôle de soutien dans la découverte bipolaire scientifique exige un tel but unique.

Il est clair que cette analyse de ce que devrait être la machine représentationnelle est orientée, et met l'accent sur des aspects de la découverte scientifique qui correspondent aux caractéristiques de l'agence. Nous aurons à revenir sur le caractère orienté de l'analyse du contenu de la machine représentationnelle. Sur base de cette analyse donc, l'agence, poursuit l'équipe de Somalvico, semble un concept adéquat pour poursuivre l'analyse de ce rôle représentationnel. Si nous faisons correspondre à la notion de modèle coopérant celle d'un agent possédant à la fois ce modèle et une capacité de coopération, nous voyons que l'agence répond aux trois exigences de la machine représentationnelle :

- l'agence est, par définition, un système multi-agents, capable de faire coopérer les modèles entre eux. Ceci remplit la première exigence.
- les entités de l'agence sont des agents qui ont une capacité inférentielle

ou non élémentaire. Ceci remplit la seconde exigence.

- l'agence exige un but unique défini avant la construction de l'agence (approche «top-down»), capable de soutenir le travail de l'homme dans le cadre de la découverte scientifique bipolaire. Ceci remplit la troisième exigence.

On voit que l'agence, dans sa définition même, est un concept adéquat pour la machine représentationnelle telle qu'elle est vue par l'équipe de Somalvico. Ils identifient donc agence et machine représentationnelle, et c'est ce qui sera supposé implicitement dans les développements qui suivent.

L'interaction entre les deux rôles : la circularité

L'équipe de Somalvico va alors mettre en évidence une propriété importante de l'agence et de ses deux rôles sous-jacents. Comme nous l'avons vu, l'agence est à même de réaliser les deux rôles que nous venons d'exposer, et de ce fait développe une interaction synergique. Le premier rôle descriptif aide le second rôle de soutien, et vice versa. Une telle interaction mutuelle est appelée *circularité*.

D'abord le rôle de description aide celui de soutien à la découverte scientifique. En effet, lorsqu'un nouveau modèle est inséré dans l'agence servant de description, l'agence renforce et améliore son fonctionnement quant au phénomène visé. Ensuite lorsque son rôle de soutien produit un nouveau modèle, ce nouveau modèle peut être incorporé à l'agence en tant qu'outil de représentation. Cette double interaction offre ainsi une circularité entre les deux rôles.

Cette double interaction amplifie l'acte de critique au sein du schéma de Galilée, et renforce par là la création scientifique. Rappelons que la critique est un acte créatif au travers duquel l'homme améliore un cycle d'abduction/adduction par rapport à un cycle précédent. Or un nouveau modèle inséré dans l'agence renforce le rôle de soutien et améliore par là le cycle de découverte par rapport au précédent. Cette amélioration au moyen de la critique peut être réitéré autant de fois que nécessaire et accentue l'efficacité de la découverte scientifique de l'homme.

On comprend que les deux rôles de la machine peuvent avoir une interaction synergique, au moyen de la circularité.

4.3 Architecture pour la description de la découverte scientifique

Si d'un point de vue des concepts, il semble que l'agence puisse remplir le rôle de la machine représentationnelle, il reste à montrer des mécanismes précis permettant à l'agence de remplir ce rôle. C'est le travail de l'équipe de Somalvico dans un article en particulier dans leur article «Dynamic Agencies and Creativity» [10].

4.3.1 La métaphore de la mosaïque

Pour pouvoir articuler le résultat de l'abduction (un modèle), note l'article, il faut disposer d'une représentation des liens entre les modèles issus des phénomènes. La figure 4.2 propose une métaphore qui détaille le rapport entre phénomène et modèles.

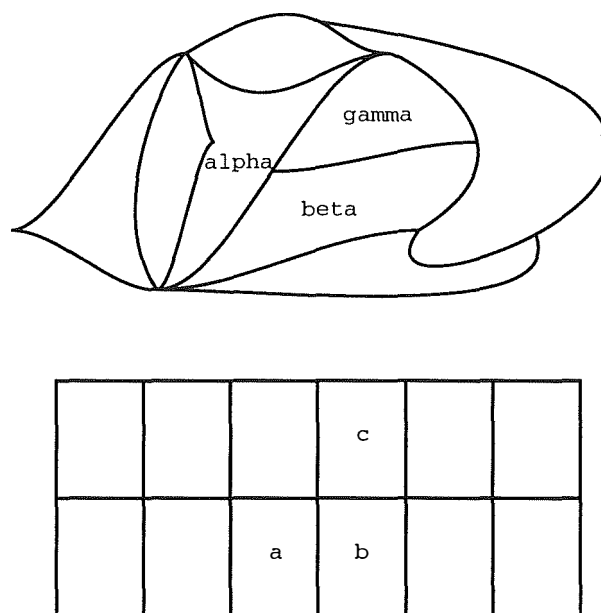


FIG. 4.2 – En haut le phénomène représenté comme un assemblage de champs, en bas un assemblage de modèles représentés comme des tuiles d'une mosaïque.

En haut de la figure 4.2, nous trouvons l'objet de l'abduction, un ensemble de phénomènes alpha, bêta, et gamma. Notons que ces phénomènes peuvent être considérés comme des côtés d'un même phénomène ou comme des phénomènes différents ayant un rapport entre eux. Ce rapport est sym-

bolisé par le bord des phénomènes alpha, bêta et gamma. Par exemple, le phénomène bêta est en lien avec le phénomène alpha et le phénomène gamma. Les champs des phénomènes ont une forme vague pour essayer de symboliser la difficulté contenue dans l'acte d'abduction. En bas de la figure 4.2, nous trouvons le résultat de l'abduction. Chaque modèle est vu comme une tuile dans une mosaïque. Chaque tuile correspond à un phénomène. Dans ce cas précis, les tuiles a, b, et c correspondent respectivement aux phénomènes alpha, bêta, et gamma. Les relations entre les phénomènes alpha, bêta et gamma sont conservés et ces relations sont symbolisées par les bords.

L'équipe de Somalvico a ainsi mis au point une représentation du point de départ et d'arrivée de l'abduction, jetant les bases d'une machine de représentation telle qu'ils l'ont pensée.

4.3.2 Agence représentationnelle

A ce point du développement, l'équipe de Somalvico va utiliser le concept d'agence pour enrichir et préciser la représentation du résultat de la créativité sous forme de tuiles.

Ils vont tout d'abord faire correspondre chaque modèle à un agent.

Un agent est toujours structuré en deux parties, la partie opérationnelle *so semiagent* et la partie coopérationnelle *co semiagent*. La partie opérationnelle correspond au modèle et à l'utilisation de ce modèle, avec toute la gestion sous-jacente. La partie coopérationnelle est la partie de communication, qui s'occupe de relier un agent à tous les autres agents, lui permettant de s'intégrer activement dans un processus de coopération entre différents agents. L'origine d'une telle division d'un agent est celle de l'aporie de la tour de Babel, que nous avons déjà abordé dans le chapitre 1 (p. 19).

Dans notre cadre, la partie *so semiagent* correspond au modèle lui-même, et la partie *co semiagent* correspond aux liens entre les agents. A chaque tuile de la mosaïque correspond un agent au sein de l'agence, et les relations entre les tuiles sont maintenues dans l'agence par les relations entre les agents.

Bien que l'acte d'abduction, poursuit l'équipe de Somalvico, soit un acte créatif non formalisable, il est possible de se le représenter en deux temps : d'abord un premier temps qui consiste en la création d'un nouveau modèle, ensuite dans un second temps qui consiste en l'intégration d'un nouveau modèle.

La création d'un nouveau modèle correspond, dans notre cadre, à la création d'une nouvelle tuile. Ce type de tuile sera représenté sous la forme d'une tuile noire, parce le modèle qu'elle contient est le fruit d'un acte créatif informalisable. Par la création d'une tuile noire, nous pouvons distinguer deux nouveaux types de tuiles : une tuile dont le modèle a un lien avec

le nouveau modèle, et une tuile dont le modèle n'a aucun rapport avec le nouveau modèle. Le premier type de tuile est appelé tuile grise, le second type tuile blanche (voir figure 4.3).

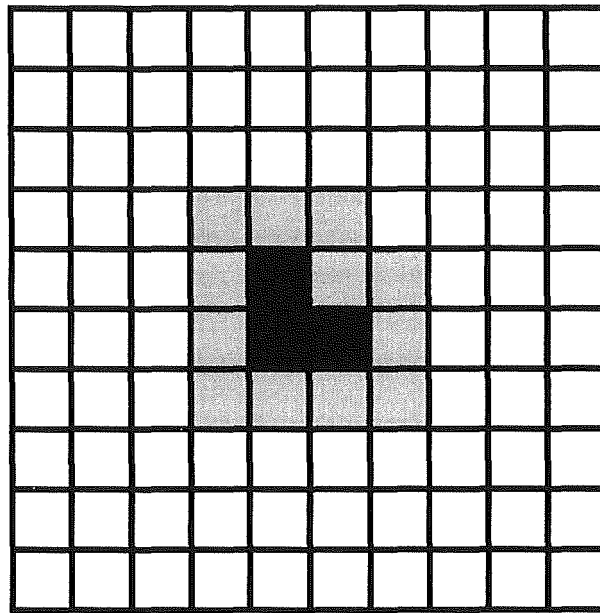


FIG. 4.3 – Les tuiles noires sont des nouveaux modèles ayant un lien avec d'anciens modèles, les tuiles grises sont des anciens modèles ayant un lien avec les nouveaux modèles, et les tuiles blanches sont des modèles anciens sans lien avec les nouveaux modèles.

L'intégration d'un nouveau modèle est la phase pendant laquelle un nouveau modèle est inséré au sein de l'agence. Lorsque l'agent représentant le nouveau modèle est inséré au sein de l'agence, les relations entre ce nouveau modèle et les anciens modèles doivent être spécifiées. L'existence de ce lien est encore un acte créatif, et il appartient donc à l'homme de les préciser.

La possibilité de créer de nouvelles relations entre les modèles provient de la caractéristique de l'agence de séparer un agent en deux parties. Grâce à cette séparation, un modèle qui correspond à la partie fonctionnelle de l'agent peut avoir une nouvelle relation avec les autres modèles, en modifiant uniquement la partie coopérationnelle.

Les agents représentant les tuiles auront des parties opérationnelles et coopérationnelles différentes. Un agent représentant une tuile noire aura un nouveau so semiagent et un nouveau co semiagent. Un agent représentant une tuile grise aura un ancien so semiagent et un nouveau co semiagent. Enfin, un agent représentant une tuile blanche aura un ancien so semiagent

et un ancien co semiagent. On voit tout l'intérêt d'une découpe des agents en deux parties, l'une opérationnelle, l'autre coopérationnelle.

En conclusion, l'équipe de Somalvico a précisé la représentation des liens entre les différents modèles, et a également donné la manière dont de tels liens, avec les modèles sous-jacents, pourraient être implémentés au sein de l'agence par le biais des agents. L'intervention de l'homme, en vertu de son intelligence créatrice, se situe à deux niveaux : d'abord en tant que créateur de modèle, ensuite en tant que créateur des liens qui unissent les modèles aux anciens.

Une agence qui respecterait les principes que nous venons d'exposer se nomme *agence représentationnelle*.

4.3.3 Différents niveaux d'abstraction

Jusqu'à présent, l'agence est capable de représenter les liens entre différents modèles pour *un et un seul* phénomène. Ce premier type de relation entre les modèles peut être appelé relation d'interaction entre les modèles. C'est un premier type de lien minimal entre les modèles en vue de représenter le savoir scientifique : le savoir scientifique lie parfois des phénomènes entre eux, qu'ils soient des phénomènes différents ou qu'ils soient des phénomènes inclus dans d'autres phénomènes.

L'équipe de Somalvico va mettre au point un second type de lien entre les modèles, en introduisant des niveaux d'abstraction dans la représentation du savoir scientifique.

Commençons par réfléchir en termes de phénomènes et de modèles. Un phénomène peut être dit un sous-phénomène d'un autre phénomène si le phénomène implique nécessairement la survenance de ce sous-phénomène. En termes de champs, cela signifie qu'il existe dans un champ donné un ou plusieurs sous-champs. Une telle relation de composition intervient lorsque l'homme considère un phénomène à différents niveaux d'abstraction. Par exemple, on peut modéliser le phénomène du feu, mais on peut aussi modéliser le phénomène de la consommation d'oxygène sous-jacent. On peut alors considérer une relation de composition entre un champ et ses sous-champs. Le champ est dit composé et un sous-champ composant. Le champ est un niveau d'abstraction plus haut que le sous-champ.

Nous pouvons faire correspondre à ce schéma une structure de modèles en tuiles. Au phénomène correspond une tuile, éventuellement liée à d'autres tuiles au même niveau d'abstraction. Une tuile peut elle-même être composée de sous-tuiles qui correspondent aux sous-champs du phénomène. Au niveau inférieur, une sous-tuile entretient des relations avec les autres sous-tuiles comme pour le niveau supérieur. Bien entendu, une sous-tuile peut elle-même

être composée d'autre tuiles. On parle alors de mosaïque et de sous-mosaïque.

Ajoutons que les sous-tuiles de différents phénomènes ne sont pas spécialement disjoints. De plus, un même phénomène peut être décomposé en des ensembles différents de sous-tuiles.

Nous possédons ainsi une structure d'enchevêtrement de tuiles et de sous-tuiles qui représente différents niveaux d'abstraction d'un phénomène.

Soulignons que les deux relations d'interaction (à un niveau d'abstraction) et de composition (entre deux niveaux d'interaction) ne sont pas spécifiées et sont supposés implémentées dans la partie coopérative des agents.

Nous abordons à présent la question de savoir comment intégrer de tels niveaux d'abstraction au sein d'une agence en général. Ceci est possible grâce à la propriété d'échelle intrinsèque à l'agence (voir chapitre 1, page 17) : une agence doit être considérée comme une machine unitaire, c'est-à-dire réalisant un objectif précis, et dès lors une agence peut être considérée à son tour comme un agent. Si nous nommons une sub-agence une agence représentant un niveau inférieur d'abstraction, on peut alors affirmer qu'une sub-agence est un agent unique, si bien qu'elle peut être intégrée dans une agence, au sein d'autres sub-agences. Ce concept d'échelle est évidemment récursif, et il est clair que l'on obtient une structure enchevêtrée semblable à celle de la mosaïque à multiples niveaux. L'agence représentationnelle nous offre une fois de plus l'avantage de rendre possible une telle relation puisque les parties coopératives, c'est-à-dire les relations, sont reprogrammables et l'intégration d'un agent idéal qui est une sous-agence est alors toujours réalisable.

L'équipe de Somalvico a ainsi élaboré une agence représentationnelle à plusieurs niveaux d'abstraction.

4.3.4 Innovation dans la découverte scientifique

L'équipe de Somalvico a jusqu'à présent donné une représentation statique de la représentation du savoir scientifique. Il faut prendre en compte l'insertion d'un nouveau modèle au sein de cette représentation, c'est-à-dire qu'il faut prendre en compte l'innovation. Plus précisément et conformément au schéma donné dans la section 4.3.2 (p. 79), il s'agit de détailler le processus impliqué lorsqu'un nouveau modèle (une tuile noire) est inclus au sein d'anciens modèles (une tuile blanche) et crée ainsi des liens avec d'autres modèles (une tuile grise). Nous séparerons cette description en deux parties : d'abord l'innovation au même niveau d'abstraction, ensuite à des niveaux d'abstraction différents.

Au même niveau d'abstraction

Considérons d'abord un niveau donné. On peut différencier six types d'innovation, représenté par la figure 4.4.

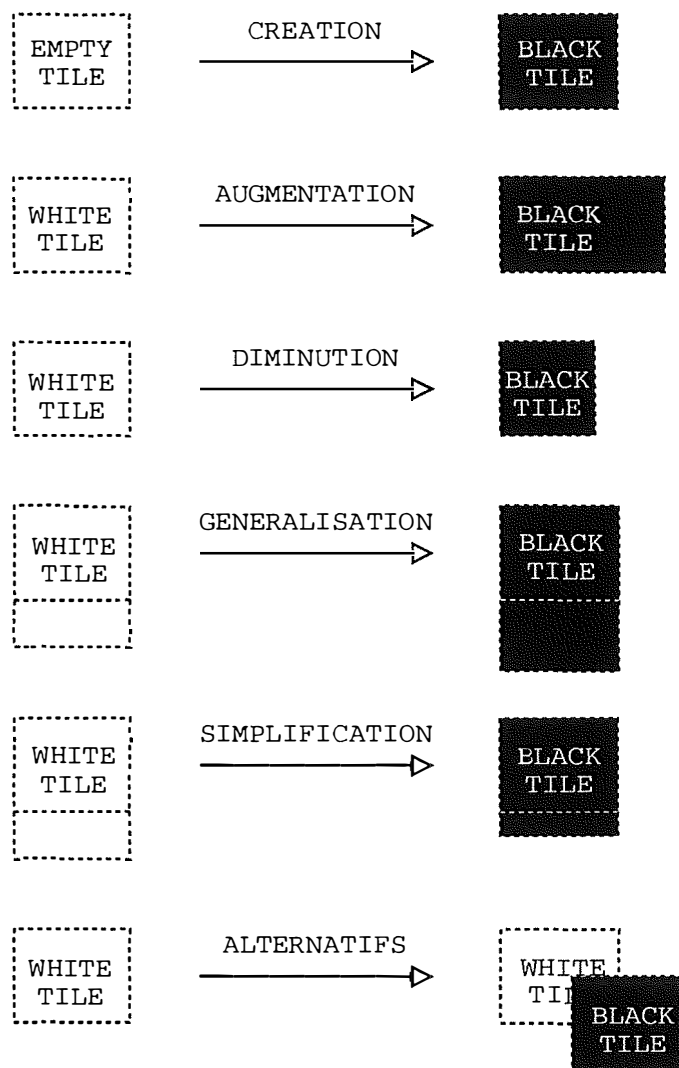


FIG. 4.4 – Les différentes méthodes pour insérer un nouveau modèle (tuiles noires) dans une mosaïque.

Le cas le plus simple est celui de la création, où un modèle d'un phénomène est inséré mais pour lequel aucun autre modèle n'a jamais été créé. Les autres méthodes d'insertion insèrent un modèle pour lequel un modèle plus vieux existe déjà.

Pour pouvoir parler par exemple de simplification et de généralisation en ce qui concerne un modèle, il nous faut une mesure de la précision de ce modèle. Pour ce faire, on peut remarquer d'une manière générale qu'un modèle est composé d'une forme et d'un formalisme. La forme est la partie du phénomène que l'on choisit de modéliser, le formalisme est l'ensemble des moyens utilisés pour représenter la forme choisie. La forme représente donc l'abstraction effectuée sur le réel, c'est-à-dire le choix des éléments à retenir dans le phénomène. L'acte de modélisation, de ce point de vue, peut être décrit comme l'acte de projeter la forme d'un phénomène dans un formalisme (voir section 4.2.2).

L'équipe de Somalvico propose alors de mesurer la complexité d'un modèle au moyen du nombre de symboles que contient le formalisme. Le nombre de symboles que contient un formalisme est en effet lié à la richesse et la précision du modèle. De plus il faut ajouter la contrainte que les formalismes se trouvent sous une forme canonique, c'est-à-dire que tout symbole ne portant pas de véritable information doit être simplifié. Bien entendu il reste à savoir si ce critère est effectivement un critère adéquat pour évaluer la richesse d'un modèle scientifique. Nous reviendrons sur ce point dans le chapitre 6.

Nous considérons à présent le premier cas où seule la forme varie, mais pas le formalisme. On peut mettre en évidence diverses méthodes d'insertion :

- Augmentation. Le nouveau modèle modifie la forme de l'ancien modèle en augmentant le nombre de symboles. Le nouveau modèle décrit donc de manière plus précise le phénomène.

- Diminution. Le nouveau modèle modifie la forme de l'ancien modèle en diminuant le nombre de symboles. Le nouveau modèle décrit de manière moins précise le phénomène.

- Généralisation. Le nouveau modèle substitue une partie des symboles de l'ancien modèle avec une partie plus riche, c'est-à-dire contenant plus de symboles. Le nouveau modèle décrit de manière plus précise que l'ancien modèle.

- Simplification. Le nouveau modèle substitue une partie des symboles de l'ancien modèle avec une partie moins riche, c'est-à-dire contenant moins de symboles. Le nouveau modèle décrit de manière moins précise que l'ancien modèle.

Il est clair que ces méthodes ne sont pas indépendantes. D'une part la généralisation et la simplification peuvent être considéré comme des applica-

tions des deux méthodes plus élémentaires que sont la diminution et l'augmentation. D'autre part l'augmentation peut être considérée comme un cas particulier de la généralisation, en forçant la substitution de la généralisation à être vide. La diminution peut aussi être considérée comme un cas particulier de la simplification, en forçant la substitution à être vide. La volonté de distinguer ces quatre méthodes relève plutôt de leur puissance cognitive, c'est-à-dire de leur utilité lorsqu'il s'agit de représenter l'innovation au sein de la représentation du savoir scientifique. Nous pouvons résumer ces méthodes en disant que l'augmentation et la généralisation augmentent le nombre de symboles par rapport à l'ancien modèle, et que la diminution et la simplification diminuent le nombre de symboles par rapport à l'ancien modèle.

Nous levons maintenant l'hypothèse selon laquelle le formalisme est identique dans tous les modèles. Commençons par considérer le cas où les formalismes des deux modèles ont un rapport hiérarchique. Nous avons pour cela besoin de définir une notion d'abstraction. Nous dirons qu'un formalisme alpha est plus abstrait qu'un formalisme bêta lorsque tout symbole de alpha peut être représenté comme une composition de symboles de bêta, mais pas l'inverse. Si le formalisme de l'ancien modèle est plus abstrait que celui du nouveau modèle, l'insertion correspond alors à une simplification. On entend bien par simplification le fait de laisser certains détails de côtés en vue de retenir l'essentiel. Si le formalisme de l'ancien modèle est moins abstrait que celui du nouveau modèle, l'innovation consiste alors en une généralisation, puisque le nouveau modèle contient l'ancien.

Le cas où les formalismes n'ont aucun rapport hiérarchique nous amène à considérer la dernière méthode de la figure 4.4. Si les formalismes n'ont aucun rapport hiérarchique entre eux, alors les modèles ne sont pas commensurables. Cela signifie qu'ils sont deux modèles alternatifs à un même phénomène. Par exemple, la lumière peut être décrite selon deux modèles en physique, d'abord comme une onde électromagnétique ensuite comme un ensemble de protons.

Il reste enfin un cas fort intéressant qui n'implique aucune insertion de modèle mais qui consiste quand même en une innovation, mais une innovation «faible». Il s'agit de la réorganisation du savoir, soit que cette représentation soit plus claire soit que cette représentation mette en évidence certaines relations. En termes de tuiles cela signifie que les bords des tuiles sont mises en relation avec de nouvelles tuiles. En terme d'agent, cela signifie que la partie so semiagent de tous les agents est conservée, mais que la partie co semiagent est modifiée pour établir de nouvelles relations entre les modèles.

A différents niveaux d'abstraction

L'équipe de Somalvico a jusqu'à présent considéré les différents moyens d'innovation à un seul niveau donné. Il s'agit à présent d'étudier l'insertion d'un nouveau modèle à différents niveaux d'abstraction, c'est-à-dire sur la relation de composition. A cet égard il est important de noter que nous ne précisons pas la nature des relations entre les modèles à un niveau donné et entre les modèles à différents niveaux, qui seront inclus dans la partie co-semiagent des agents. La spécification de ces liens est un acte créatif qui doit être effectué par l'homme. Rappelons également que chaque modèle est lui-même décrit par des sous-modèles, c'est-à-dire que le phénomène du modèle est composé d'autres phénomènes à un niveau d'abstraction inférieur ; de plus, chaque modèle fait lui-même partie d'un modèle composé, c'est-à-dire que le phénomène du modèle compose un autre phénomène. Chaque relation est appelée, respectivement, une relation de composition de niveau inférieur et une relation de composition de niveau supérieur.

Nous commençons par décrire la relation de composition de niveau inférieur. Considérons un modèle **a** à un certain niveau et les modèles **a1**, ..., **an** à un niveau d'abstraction inférieur. Nous pouvons distinguer deux types d'association. Le premier type est l'*association faible*, lorsque les sous-phénomènes représentés par les modèles **a1**, ..., **an** ne recouvrent pas le phénomène représenté par le modèle **a**. Le second type est l'*association forte*, lorsque les sous-phénomènes représentés par les modèles **a1**, ..., **an** recouvrent totalement le phénomène représenté par le modèle **a**.

De manière analogue, nous pouvons préciser la relation de composition de niveau supérieur en considérant un modèle **a** à un certain niveau et les modèles **a1**, ..., **an** à un niveau d'abstraction supérieur. La situation est similaire à la relation de composition de niveau inférieur, et nous retrouvons les notions d'association faible et d'association forte. Remarquons néanmoins qu'un phénomène décrit par un modèle peut être un des sous-phénomènes de plusieurs modèles de niveau supérieur. Ainsi un modèle **a** est en association forte avec les modèles **a1**, ..., **an** de niveau supérieur si le modèle **a** compose chacun des modèles **a1**, ..., **an**. Nous parlerons d'association faible dans le cas contraire.

4.4 Conclusion

L'équipe de Somalvico, sur base de leur cadre épistémologique, dégage deux rôles fondamentaux pour la machine au sein de la découverte scientifique : le soutien et la description. Somalvico et son équipe considèrent que

ce soutien sera toujours partiel, c'est-à-dire que la machine est capable de simuler seulement l'intelligence fabricatrice. Devant cette dualité entre intelligence créatrice et fabricatrice, surgit le concept de découverte scientifique bipolaire. Les deux rôles de la machine trouvent de plus leur unification au sein d'une machine qui serait capable d'intégrer ces deux pôles en elle-même, qui impliquerait alors une propriété de circularité entre les deux rôles : la création de nouveaux modèles sont intégrés au sein de la représentation de la machine, et la représentation de la machine effectue le soutien à l'activité scientifique. La circularité des deux rôles renforce ainsi la découverte scientifique.

L'agence est vue par l'équipe de Somalvico comme un candidat sérieux car elle est capable de régir l'interaction entre des modèles en vue d'atteindre un objectif. Elle remplit le rôle de soutien. Par la médiation de la métaphore de la mosaïque, il est possible d'obtenir une architecture pour la description du savoir scientifique au sein de l'agence elle-même. Le nouveau paradigme dans lequel est né l'agence, qui est celui de l'interaction entre des entités non élémentaires, rend possible la représentation opérationnelle du savoir en faisant coopérer un ensemble de modèles.

Chapitre 5

Le concept de représentation

5.1 Introduction

Nous proposons dans ce chapitre d'étudier la représentation en intelligence artificielle. De quelle manière l'intelligence artificielle voit-elle son objet ? Cette question importante aura des conséquences directes sur la modélisation et en particulier sur le générateur de modèle que nous avons abordé dans le chapitre 3. La représentation en intelligence artificielle prend sa source dans la création de la machine universelle de Turing, et en particulier dans les thèses de Church et de Turing. Après avoir défini d'une manière générale la notion de représentation, nous montrerons comment les thèses de Church et de Turing donnent naissance à un cadre d'interprétation qui engendre trois registres : le formel, l'intuitif, et le physique.

5.2 La représentation

Le concept de représentation tel que nous l'abordons dans cette section est compris sous sa signification la plus générale : il ne s'agit pas de comprendre ce qui peut être désigné comme une représentation à l'intérieur d'un système donné, mais de comprendre ce concept dans le cadre de la théorie de la connaissance.

Nous nous baserons sur les travaux de Ladrière [14, 15] qui a tenté de circonscrire ce concept non seulement dans l'histoire de la philosophie, mais également dans le champ de l'intelligence artificielle.

La notion de représentation repose, en première analyse, sur une double métaphore, celle de la « mise en présence » et celle de la « diplomatie ». La première métaphore insiste sur le fait que la représentation rend présent ce qui ne l'est pas directement. La représentation, par essence, évoque quelque

chose d'autre qu'elle-même. Elle a un pouvoir symbolique. Certains sigles religieux, par exemple, évoquent parfois des civilisations entières. La seconde métaphore relève de la diplomatie : une personne peut parfois prendre la place d'une autre, la représenter, et jouir par là des mêmes pouvoirs. Dans ce cas, il y a un transfert d'attribution entre la personne présente et la personne absente. La première métaphore met l'accent sur la mise en présence, qui relève du pouvoir symbolique, et la seconde métaphore met l'accent sur le fait d'«être à la place de», qui relève du transfert d'attribution. Les deux métaphores sont en fait liées : d'un côté, si la représentation rend présent la chose représentée, c'est que la représentation prend nécessairement la place de la chose représentée ; dans l'autre sens, la représentation ne peut prendre la place de la chose représentée que si celle-ci est mise en présence au moyen du pouvoir symbolique de la représentation.

Dans la représentation nous observons toujours deux présences : d'abord la présence de l'objet représentant, ensuite la présence de l'objet représenté, mais médiatisé par l'objet représentant. La représentation fait de l'objet représentant un pur instrument qui vise une autre présence.

Dans ce contexte, la représentation joue un rôle crucial dans le domaine de la connaissance. L'acte de connaissance se divise classiquement en deux pôles : le pôle sujet et le pôle objet. Or, par le biais de la représentation, le sujet met en présence dans sa subjectivité l'objet, par le moyen d'une médiation produite par la conscience, c'est-à-dire la représentation elle-même, tout en laissant cet objet dans son pôle objectif, à jamais hors de la sphère subjective. L'objet étudié doit prendre place dans la vie subjective, c'est-à-dire que le sujet doit pouvoir se le mettre en présence, dans le sens de la première métaphore. Mais comme l'objet réel ne peut venir se placer lui-même dans la conscience du sujet, il faut qu'il y ait une médiation : c'est le sens de la seconde métaphore.

La représentation est la méditation de la conscience rendant présent une réalité qui n'est pas accessible.

Toute la question, qui sera également centrale pour notre recherche, est de savoir comment cette médiation peut renvoyer adéquatement à l'objet que vise l'acte de la connaissance.

5.3 Le rôle herméneutique de la notion de représentation

Nous avons évoqué le rôle de la notion de représentation, comme médiation nécessaire entre le sujet et l'objet. Intéressons-nous à présent à la

nature de cette médiation lorsque l'acte de connaissance a pour objectif la simulation sur ordinateur. Dans son article «Note sur l'emploi et l'origine de la notion de représentation en sciences cognitives», Jean Lassègue [16] essaye de montrer que cette notion de représentation a surtout un rôle herméneutique, c'est-à-dire qu'elle définit des dimensions intuitives, formelles, et physiques et qu'elle affirme des liens idéaux entre ces dimensions. La réflexion de Lassègue porte plus spécifiquement sur les sciences cognitives, mais nous pouvons prendre appui sur cette réflexion, puisque la réflexion que mène Lassègue reste valable tant que la représentation est analysée en lien avec l'utilisation de l'ordinateur. Dans un premier temps, nous essayerons de montrer, avec Lassègue, l'origine contemporaine de la théorie de la représentation. Ensuite, dans un second temps, nous essayerons de montrer l'origine philosophique de cette même théorie, en particulier en remontant jusqu'à la conception cartésienne de la notion de représentation.

5.3.1 La genèse de la théorie de la représentation

Lassègue [16] fait remonter la genèse de la théorie de la représentation à Church et à Turing ¹.

La thèse de Church

Church, dans son article «A note on the Entscheidungsproblem» [1], pose la question de savoir s'il existe un algorithme capable de décider pour tout problème de sa déductibilité. Mais l'on tombe dans une aporie. D'une part il faudrait prouver algorithmiquement que toute opération de l'esprit peut être incluse dans la notion d'algorithme, mais, d'autre part, la notion d'algorithme est intuitive.

La notion d'algorithme semble devoir être définie formellement, de telle sorte que tout ce qui est intuitivement compris dans la notion informelle d'algorithme se trouve dans la nouvelle notion formelle. Church exhibe alors plusieurs notions formelles équivalentes entre elles, qu'il rapporte à la notion d'algorithme. Church tente donc de justifier l'équivalence entre opération de l'esprit et la notion informelle d'algorithme en montrant l'équivalence de notions formelles censées recouvrir la notion intuitive d'algorithme. On passe de l'affirmation d'un terme (la notion informelle d'algorithme) à un rapport (les équivalences entre les notions formelles).

¹Pour un exposé de la thèse Church-Turing, voir l'encyclopédie de philosophie en ligne de Standford : B. Jack Copeland, «The Church-Turing Thesis», *The Stanford Encyclopedia of Philosophy (Summer 2002)*, Edward N. Zalta (ed.), URL = <http://www.science.uva.nl/seop/archives/sum2002/entries/church-turing/>

En fait, l'identification opérée par Church est informelle et montre que cette correspondance entre la notion d'algorithme et les notions formelles qui s'y rapportent doit être continuellement vérifiée par le sujet. On atteint donc là une limitation des pouvoirs mathématiques des êtres humains.

Derrière la thèse de Church, on voit se profiler le registre de l'intuition au sein de la notion de calcul, puisqu'il semble que la notion intuitive d'algorithme soit sans cesse au-delà de toute formalisation, au point que le sujet doive prouver continuellement cette identification.

Turing va tenter d'apporter de nouveaux arguments informels pour appuyer cette identification, au moyen de la *machine universelle*.

La thèse de Turing n° 1

Dans «On Computable Numbers with an Application to the Entscheidungsproblem» [2], Turing analyse les étapes de l'être humain en train de calculer et effectue une correspondance avec une machine abstraite, qui fonctionne discrètement et procède par états. La machine abstraite est capable d'imiter l'homme dans le domaine du calcul, elle *représente* donc l'homme en train de calculer.

Derrière cette première thèse de Turing, on voit surgir un second registre au sein du calcul : le calcul est fondamentalement discret, c'est-à-dire qu'il peut être ramené à une machine abstraite fonctionnant discrètement par états.

La thèse de Turing n°2

Dans son article «Computing Machinery and Intelligence» [3], Turing tente d'effectuer une identification *physique* entre la machine universelle et un être humain en train de calculer.

Pour ce faire, Turing remarque que l'être humain calcule discrètement comme la machine universelle, et qu'aussi bien l'être humain que la machine universelle fonctionnent physiquement de manière *continue*.

On trouve alors une analogie entre l'être humain et la machine universelle, aussi bien au niveau du calcul qu'au niveau physique.

Sur base de cette analogie, on peut espérer concevoir une science de l'esprit qui aurait comme moyen le calcul et comme outil la machine universelle.

Au terme de l'analyse des thèses de Church et de Turing, Lassègue souligne l'apparition de trois registres, nécessairement induits par le travail

de Church et de Turing :

- un registre intuitif, apparaissant dans la première thèse de Church,
- un registre formel, avec la première thèse de Turing qui tente de formaliser le calcul humain en un calcul discret,
- un registre physique, avec la seconde thèse de Turing qui insiste sur la constitution physique continue de l'être humain et de la machine universelle.

Comment ces registres sont-ils liés, et qu'est-ce qui les caractérise ?

Le premier registre est celui du sujet qui doit continuellement confirmer l'identification non formellement fondée de la thèse de Church et par conséquent des deux thèses de Turing ; cette tâche est donc infinie, d'autant plus que le domaine auquel s'applique le calculable est également infini. Le second registre met à jour le principe du *discret*, par le biais du calcul formel réalisé par la machine universelle ; le discret est dès lors le principe qui trace la limite entre ce qui est représentable (ce qui peut être ramené à du récursif d'un point de vue discret et qui s'exprime dans la thèse de Church) et ce qui est non représentable (ce qui est de l'ordre du continu, et qui s'exprime dans la seconde thèse de Turing). Turing, dans sa seconde thèse, inaugure un troisième registre, en regardant la nature comme continue, et donne un statut à ce qui est au-delà du discret. Cet excès de la représentation est néanmoins *pensable*, c'est-à-dire que le sujet, par le biais de la tâche infinie de réappropriation, est conscient de ce qui n'est pas formalisé.

On peut donc caractériser les trois registres de l'intuitif, du formel et du physique par l'infini, le discret et le continu respectivement. Leur liens sont des liens de dépassement, qui se manifestent comme ce qui dépasse le calculable.

Le modèle de l'ordinateur implique que toute représentation soit pensée sous ces trois registres, avec les liens qui les unissent. La notion de représentation génère d'elle-même un cadre herméneutique, c'est-à-dire qu'elle fixe l'interprétation de l'objet des sciences cognitives. Le calculable est identifié, mais où est en même temps et indissociablement associé le dépassement de ce calculable. La représentation implique l'ordre du discret, par le biais du modèle de l'ordinateur, mais trace en même temps les limites de ce discret.

5.3.2 Les origines cartésiennes de la notion de représentation

Lassègue insiste sur le fait que cette structuration en trois registres générée par la notion de représentation ne date pas des thèses de Church et Turing. Autrement dit, cette théorie de la représentation ne date pas du paradigme de l'ordinateur. On peut en effet faire remonter cette structuration à Descartes, qui a déjà inauguré ces trois registres.

Une représentation, pour Descartes, est une idée présentée à l'esprit dont on peut retracer la *construction* [16, p.6]. Dit négativement, toute idée que l'on ne peut pas reconstruire n'est pas une représentation, ou n'est pas une idée claire et distincte, pour reprendre le langage de Descartes. La construction est au centre de la représentation, et cette représentation est de l'ordre du formel, car la construction possède un caractère discret.

Mais le formel est dépassé par le physique. En effet, la notion de construction possède comme condition de possibilité le mouvement physique. D'une part, le mouvement *échappe* à toute construction parce que ce mouvement ne peut être capturé par une mesure ; d'autre part, le mouvement *permet* la construction, en passant, par exemple en géométrie, d'un point à l'autre.

Du point de vue de l'esprit, le mouvement se manifeste comme inaccessible à la construction. Mais si l'esprit est capable de mettre en correspondance le continu et le discret, c'est qu'il possède intuitivement le mode du continu, alors qu'il ne peut construire ce continu. La notion de direction manifeste à l'esprit cette dualité : la direction est en effet un mouvement, puisqu'il s'agit de l'indication d'un chemin potentiel toujours prolongeable, et la direction possède pourtant en elle-même son principe de construction. Si l'on considère le point de vue de l'esprit, on a donc deux conséquences. D'une part le mouvement continu de l'esprit est inaccessible à toute construction, limitation qui se manifeste dans la notion de direction. D'autre part, comme on l'a montré plus haut, ce mouvement continu est plutôt la condition de possibilité de la construction.

Du point de vue de la machine, le même rapport entre mouvement et construction se révèle : c'est en effet le mouvement continu qui permet toute construction au moyen de la machine.

Nous trouvons donc un rapport de dépassement entre le formel et le physique. L'analogie entre la machine et l'esprit consiste en ce que le mouvement continu est la condition de possibilité de la construction.

Comme chez Turing, on retrouve une même analogie entre machine et esprit et un même rapport de dépassement entre le niveau du formel caractérisé par le discret et le niveau du physique caractérisé par le continu.

Le rapport entre le formel et l'intuitif est également un rapport de

dépassement : Descartes remarque que sa machine trace plus d'objets que l'on peut en construire formellement. Dans l'esprit humain, on retrouve bien sûr l'intuition du dépassement du formel, par exemple dans le langage poétique.

Remarquons enfin que le physique et l'intuitif jouent des rôles analogues : le physique joue le rôle du dépassement dans le cas de la machine, et l'intuitif joue ce rôle dans le cas de l'esprit.

La pensée de Descartes semble donc bien donner naissance à ces différents registres, registres que nous avons retrouvé chez Church et Turing.

5.3.3 Conclusion : la représentation dans le paradigme de l'ordinateur

La représentation, liée à l'utilisation de l'ordinateur, implique le déploiement de trois registres. Aussi chez l'un des initiateurs de la science moderne et, de manière plus significative, chez les pères de l'informatique, on retrouve ces trois registres, l'intuitif, le formel et le physique. Le registre central, immédiatement mis en jeu dans la représentation, est celui du formel, caractérisé par le discret ; mais les registres de l'intuitif et du physique sont également présents, comme des horizons de dépassement. Il faut donc prendre conscience, à ce stade, que la représentation informatique implique constitutivement l'utilisation de ces registres.

5.4 Conclusion

Dans la théorie générale de la connaissance, la représentation est la médiation de la conscience qui vise l'objet. La question centrale est alors de savoir comment cette médiation renvoie à l'objet. Dans le cadre du paradigme de l'ordinateur, qui nous intéresse dans ce travail, la médiation représentative renvoie à l'objet au travers des trois registres de l'intuitif, du formel et du physique, caractérisés respectivement par l'infini, le discret et le continu. Il faut donc comprendre, à travers cette analyse, que dès que l'ordinateur rentre en jeu dans le processus de connaissance, il se tisse une toile qui filtre l'objet à connaître, non par la volonté arbitraire du sujet de simplifier ou de mettre l'accent sur certains de ses aspects, mais nécessairement, de par la nature même du paradigme de l'ordinateur. Autrement dit, la structuration en trois registres que nous avons expliquée dans ce chapitre est transcendantale au sujet connaissant.

Les conséquences d'un tel mode de connaissance au sein du paradigme de l'ordinateur sont nombreuses et s'étendent sur tout le champ de la

modélisation propre aux sciences informatiques, modélisation qui fera l'objet du prochain chapitre.

Chapitre 6

Le concept de modèle

6.1 Introduction

Ce chapitre a pour objectif de définir précisément la différence du rapport sujet/objet dans les sciences empiriques et en intelligence artificielle. Nous définissons d'abord la notion générale de modèle ; nous précisons alors le rôle du modèle dans les sciences empiriques, qui nous permet de comprendre les rapports entre théorie, modèle et expérimentation. C'est en s'appuyant sur cette réflexion sur les sciences empiriques que nous montrons comment en intelligence artificielle l'expérimentation prend un autre sens, en remplaçant l'extériorité de la nature par une extériorité interne de la pensée. Nous portons alors un regard critique sur les thèses entourant le concept d'agence.

6.2 La fonction du modèle

Jean Ladrière présente la notion de modèle comme un cas particulier de celle de représentation [15]. Le modèle est une médiation particulière, qui peut elle-même se particulariser. On aura par exemple des modèles schématiques, ou semi-formels, ou mathématiques. Le modèle n'est qu'une modalité de la représentation, parce qu'un signe peut tout aussi bien représenter un objet, et un simple signe (par exemple un mot) n'est généralement pas considéré comme un modèle, d'autant plus, comme on verra, qu'un mot ne peut donner lieu à la recreation de l'objet considéré.

Le concept de modèle porte également une double ambiguïté. D'une part il peut être entendu dans le sens de point de référence, comme c'est le cas pour l'artiste lorsqu'il vise à reproduire un modèle, ou celui de l'ingénieur voulant réaliser un artefact. Dans ce sens, on part d'un objet idéal en vue de le reproduire concrètement. Le mouvement est celui du modèle jusqu'à

l'objet. C'est principalement un mouvement de production. Le modèle idéal est uniquement pensé et n'a pas besoin d'être réalisé. C'est la décision de l'artiste et donc de l'homme en général de le produire, mais cette production est nécessairement une dégradation, l'objet est une image déformée du modèle idéal.

D'autre part, le modèle peut être conçu comme une image de l'objet qui lui correspond de manière approximative. La relation de similitude est toujours présente mais le mouvement, ici, est inversé, allant de l'objet jusqu'au modèle. L'objet est cette fois la réalité parfaite, le modèle en est une image plus ou moins grossière. Mais ce modèle constitue une aide pour penser l'objet.

Le premier sens est de l'ordre de l'action, car il s'agit de construire un objet, tandis que le second sens est de l'ordre de la connaissance, car le but est de connaître l'objet au moyen d'une image dégradée.

On a donc deux mouvements distincts : un mouvement descendant, du modèle à l'objet, et un mouvement ascendant, de l'objet au modèle.

Mais les deux sens du terme modèle peuvent être réunis en un seul : en fait les deux approches commencent par la conception d'un modèle avec lequel on aboutit à l'objet. Dans le premier cas, crée d'abord un modèle idéal sur lequel se baser en vue de créer l'objet ; dans le second cas, on crée d'abord un modèle approximatif en vue de mieux connaître l'objet lui-même. Il existe donc un rôle commun entre celui de création et celui de connaissance, puisque dans les deux cas on crée un modèle pour *anticiper l'objet*. Le fait que l'objectif final diffère de l'un à l'autre ne modifie pas la fonction commune du modèle qui est d'anticiper l'objet. Dans les deux cas, le modèle est la clé qui nous permet d'apprivoiser l'objet, peu importe dans quel but. Si nous rapprochons ces deux sens, c'est que nous créons le modèle comme si nous voulions le recréer, comme si nous nous mettions du point de vue d'un concepteur omniscient. Autrement dit, connaître c'est prendre la place de ce concepteur omniscient, et l'acte de connaissance est donc de l'ordre du «faire».

L'activité de l'ingénieur se décompose en deux étapes : d'abord la conception du modèle, ensuite la réalisation concrète du modèle. La première étape construit un modèle abstrait, la seconde le réalise concrètement. Or il est possible de s'arrêter à la première étape, en construisant un modèle qui *pourrait* être construit effectivement, mais à ce stade on utilise ce modèle en vue de mieux comprendre et connaître l'objet.

En conclusion, le modèle possède une fonction unique qui est l'anticipation de l'objet ; cette anticipation peut mener autant à la création réelle de cet objet, qu'à la compréhension de celui-ci. Par conséquent, la modélisation est l'acte par lequel nous tentons de nous mettre dans la position du concepteur omniscient de l'objet.

6.3 Le modèle dans les sciences empiriques

Quel rôle le modèle, compris comme recreation de l'objet à connaître, joue-t-il dans la pratique concrète des sciences empiriques? C'est à cette question que nous consacrons la présente section, en vue d'analyser les modifications qu'amène la modélisation informatique. Nous nous baserons sur le travail de Jean Ladrière [15], de Mario Bunge [19], ainsi que du chapitre 6 du livre «Le langage théologique à l'âge de la science moderne» de Jean-François Malherbe [18].

L'ensemble de ces auteurs, chacun dans leur propre perspective, insiste sur le fait que le modèle est un *intermédiaire* entre la théorie et l'expérimentation.

La théorie tout d'abord doit être vue comme un champ opératoire (intellectuel) fait d'axiomes et de règles de déduction. Ces déductions impliquent un pouvoir anticipateur. Le rôle de la théorie est principalement de prévoir les états du système mais également d'anticiper les directions de recherche qui se révéleront fructueuses.

L'expérimentation, quant à elle, est un champ opératoire (matériel). C'est grâce à des montages spécifiques et organisés que l'on effectue l'expérimentation. On opère sur la réalité au moyen d'instruments pour obtenir les données qui sont le résultat de l'expérimentation. Le rôle de l'expérimentation est de mettre à l'épreuve, au moyen de ce champ opératoire, les prédictions de la théorie.

Mais la théorie ne peut s'appliquer directement à la réalité au travers de l'expérimentation. En effet on a deux champs de nature différente, l'un intellectuel et l'autre matériel, de sorte qu'il faut un pont entre ces deux champs. La théorie est une domaine idéal face à une réalité concrète : cette réalité n'est appréhendable dans les termes théoriques que si la réalité s'y conforme. C'est précisément le rôle du modèle de faire le pont entre une réalité concrète et les concepts abstraits de la théorie. Si la théorie pouvait s'appliquer immédiatement, cela signifierait qu'elle n'effectue aucune abstraction, qu'elle ne construit aucune modèle mathématisé, qu'elle fait référence à l'objet tel qu'il apparaît; or l'une des caractéristiques essentielle de la science moderne est sa tendance à l'abstraction. Pensons par exemple à Galilée, père de la science moderne, qui effectuait des expériences seulement en pensée, alors qu'il n'avait pas les conditions expérimentales pour tester ces expériences. Ladrière par exemple insiste sur cette activité d'abstraction [15, p.438] ¹.

Si le rôle du modèle semble bien être celui de passerelle entre théorie et

¹Même si, par ailleurs, il nous faut remarquer qu'il existe certaines écoles, telles celles de l'empirisme strict, qui déclarent l'étape théorique inutile

réalité, reste à préciser le lien précis entre le modèle et ces deux termes.

Le modèle est une interprétation de la théorie. Le terme d'interprétation prend ici son sens dans le cadre des systèmes formels. Rappelons qu'une interprétation d'un système formel est un domaine d'objets concrets dans lequel il existe une correspondance entre les affirmations de la théorie et les affirmations du domaine et dans lequel il existe un critère de validité. Les affirmations déduites de la théorie doivent bien sûr être vraies dans le domaine d'interprétation. Ainsi on peut trouver différents modèles pour une théorie, c'est-à-dire différentes interprétations pour un système formel donné. On exprimera ce rapport entre théorie et modèle en disant que le modèle vérifie les axiomes et les propositions de la théorie. Dans le cadre des sciences empirico-formelles, le modèle n'est plus une «interprétation» dans le sens ci-dessus. On dira plutôt que le modèle satisfait aux exigences de la théorie, à ses conditions.

Le modèle est aussi, quand on considère son rapport à la réalité, une schématisation plus ou moins adéquate, provisoirement acceptable de la réalité. Mais cette schématisation est essentiellement une double correspondance qui s'effectue au travers de règles de correspondance : les caractéristiques du modèle sont interprétées en terme de données empiriques et inversement.

Si dans les processus de l'explication scientifique on a pu, en son temps, faire abstraction du modèle, sans doute cela était-il dû à ce que les phénomènes étaient tous observables. Il n'en va plus de même aujourd'hui où on doit récrire métaphoriquement ce qui doit être expliqué.

La conclusion de cette analyse de la pratique des sciences modernes est le rôle fondamental du concept de modèle. On peut se le représenter comme le lieu de passage nécessaire entre deux champs opératoires hétérogènes. Pour utiliser une image, le modèle est la pièce qui permet d'emboîter les deux pièces essentielles que sont la théorie et l'expérimentation. Sans modèle, la théorie ne peut être mise à l'épreuve dans l'expérimentation, et l'objet réel ne peut être relié à la théorie. La modélisation, par conséquent, est un processus de co-adaptation entre les deux champs opératoires de la théorie et de l'expérimentation.

Toute la question à ce stade est de savoir l'influence de la modélisation informatique dans ce schéma.

6.4 Sciences empiriques et ordinateur

De notre point de vue qui consiste à considérer l'ordinateur comme une machine de modélisation avant d'être un puissant outil de calcul, nous vou-

drions souligner la modification que l'ordinateur va engendrer dans le schéma de la connaissance dans les sciences empiriques. Suivant en cela Lassègue [17], nous allons souligner le déplacement en rapport avec l'expérience qui s'effectue avec l'ordinateur.

6.4.1 Spécificité de la modélisation biologique

Nous présentons à présent une réflexion de Tom Dedeurwaerde, qui s'appuie en fait sur un article célèbre de Marr. La question qui stimule la réflexion de Dedeurwaerde et de Marr est de savoir ce que sont les sciences cognitives. Ont-elles un champ propre, et en quoi se différencient-elles de la neurophysiologie, qui s'occupe également de l'étude de l'esprit humain, et de l'intelligence artificielle ? Autrement dit encore, quelle est la spécificité méthodologique des sciences cognitives ? Nous ne nous situons pas dans le domaine des sciences cognitives, mais l'articulation qui va nous être proposée permettra de mettre en évidence une méthode de modélisation en biologie, telle qu'elle est utilisée dans le cas de l'agence anthropique. Nous verrons que la spécificité des sciences cognitives se situe à un niveau computationnel, voulant comprendre la fonction de traitement d'information propre à une capacité cognitive, se différenciant de l'implémentation physique d'une telle capacité.

Computation, implémentation, et algorithmes

Pour établir des distinctions en vue de préciser la nature des sciences cognitives, nous nous référons ici au travail de T. Dedeurwaerde [25], qui se base sur l'article de Marr [23].

Dans l'étude de l'esprit, deux positions extrêmes peuvent être adoptées [25, p. 103]. D'abord une position naturaliste, se basant sur la neurophysiologie, qui affirme que les capacités cognitives doivent être réduites aux successions de mécanismes causaux biologiques. En particulier, l'enchaînement de l'activation des neurones physiques constituerait la compréhension de l'activité cognitive isolée. Ensuite, une position algorithmique, affirmant que toute opération reliant l'entrée et la sortie constitue une explication de l'activité cognitive isolée. Mais ces positions ne nous apprennent rien sur l'activité cognitive considérée [25, p. 103].

En ce qui concerne le point de vue naturaliste, on peut certes suivre le chemin des neurones activés, mais cette détection ne nous apprend rien sur l'activité considérée : nous avons un parcours des neurones, mais nous ne savons pas pourquoi ce chemin est utilisé, quelle information est utilisée, comment le système utilise cette information. On ne sait pas pourquoi les neurones empruntent ce chemin, on ne sait pas ce que le système est en train

de faire. Nous serions dans ce cas aussi ignorant que des extraterrestres ayant détecté le parcours des flux électriques au sein d'un ordinateur.

Une même objection peut être soulevée en ce qui concerne la position algorithmique. A vrai dire, l'algorithme ne nous apprend rien sur le système considéré, car à un système correspond par ailleurs une multitude d'algorithmes, et l'algorithme fournit seulement une structure logique et syntaxique qui permet de relier l'entrée à la sortie.

Ni le suivi du parcours des neurones physiques activés (neurophysiologie) ni la syntaxe des opérations reliant entrée et sortie ne permettent de comprendre ce qu'est l'activité cognitive considérée : il faut comprendre la *fonction de traitement d'information* réalisée par le système. Il s'agit de spécifier la fonction de transformation des informations présentes sur la rétine jusqu'aux représentations objectivement visibles pour nous. L'algorithme ne suffit pas, le contenu de ce que fait le système cognitif a seulement une signification par rapport à la *fonction* computationnelle. Le niveau computationnel est le niveau de cette fonction. Nous parlerons dans la suite de niveau de l'implémentational pour le niveau correspondant à l'étude physique, et de niveau algorithmique pour le niveau correspondant au développement d'algorithmes.

Le concept de computation est alors défini comme suit : "tout processus causal spécifié par une suite systématique d'opérations élémentaires de traitement de données" [25, p. 21].

Nous concluons avec Dedeurwaerde en une découpe à trois niveaux de l'étude de l'esprit, un niveau *algorithmique*, un niveau *computationnel*, et un niveau *implémentational*. (Remarquons que le niveau implémentational, contrairement au sens habituel en informatique, ne signifie *pas* l'implémentation d'un programme dans un langage particulier ; il s'agit en fait de la réalisation matérielle, qu'elle soit un ordinateur ou des réseaux physiques de neurones électroniques). Il existe un niveau spécifique computationnel, propre aux sciences cognitives, et deux autres niveaux, l'un algorithmique qu'il faut relier à l'intelligence artificielle, et l'autre implémentational qu'il faut relier à la neurophysiologie, ou à la biologie en général. La *biologie* étudie la structure matérielle, l'*intelligence artificielle* la structure syntaxique et les *sciences cognitives* la structure fonctionnelle. C'est donc dans l'étude de la fonction de traitement d'information, dans le niveau computationnel, que réside la spécificité des sciences cognitives.

Dans le point suivant, nous étudierons les liens entre les différents niveaux, espérant ainsi mettre en lumière le niveau computationnel.

Interdépendance des différents niveaux

Posons à présent la question des liens entre les trois niveaux qui émergent d'une analyse de la spécificité des sciences cognitives. Dans un premier temps, nous approfondissons le problème de l'articulation entre le niveau fonctionnel et le niveau de l'implémentation. Ensuite nous montrons, pour chaque niveau, sa dépendance par rapport aux autres niveaux.

Le problème de l'articulation entre le niveau fonctionnel et le niveau de l'implémentation biologique peut être formulé sous une nouvelle question : une réduction des niveaux fonctionnels aux niveaux physiques est-elle possible ? La réponse est non, car l'explication purement physique demeure incomplète si elle se limite aux seules causes efficientes [25, p. 113].

Prenons l'exemple de l'ADN et de l'ARN ¹. L'ADN possède une base qui est le thymine, alors que l'ARN, construit à partir de l'ADN, remplace le thymine par un uracyl. A priori, cette transposition est superflue, parce que l'ADN pourrait directement posséder l'uracyl. En fait, la fonction de l'ADN étant d'assurer l'intégrité du code génétique, l'ADN possède un thymine pour éviter toute mutation, car si l'ADN possédait un uracyl au lieu d'un thymine, cet uracyl aurait tendance à muter. Le système génétique préserve donc l'ADN de toute mutation en utilisant le thymine au lieu de l'uracyl. L'ARN n'a plus besoin d'une telle sécurité, le système génétique peut y tolérer une mutation. Par conséquent, la présence de la thymine dans l'ADN au lieu de l'uracyl s'explique seulement par l'*effet* de cette différence et non par les *causes* physiques. La chimie organique explique parfaitement cette différence entre l'ADN et l'ARN en explicitant la chaîne causale des réactions chimiques qui mènent à l'ADN et à l'ARN, mais est incapable d'expliquer la raison d'être de cette différence : pour la chimie, cette différence est un accident, mais en fait cette différence dépend de la *fonction* de l'ADN et de l'ARN. La présence de thymine s'explique par le rôle fonctionnel de l'ADN dans un système plus large ; la thymine a pour but de minimiser les mutations.

Nous devons atteindre un niveau d'explication supérieur, qui prend en compte la fonction des systèmes considérés : c'est le niveau computationnel.

Le niveau algorithmique dépend du niveau computationnel, car celui-ci donne sens aux constructions syntaxiques du niveau algorithmique. Le niveau algorithmique construit, après avoir défini le format des entrées et des sorties, un algorithme, c'est-à-dire un système de symboles, reliant entrée et sortie. Ces symboles ne prennent sens que par rapport à une fonction donnée. Le niveau computationnel donne un rôle fonctionnel aux algorithmes. De plus, une même fonction peut être calculée par un grand nombre d'algorithmes,

¹Voir Rosenberg A., *The Structure of Biological Science*, Cambridge, 1985, pp. 38-42.

et deux algorithmes peuvent recouvrir la même fonction. Il n'y a donc pas de détermination du niveau computationnel vers le niveau algorithmique ; le niveau algorithmique a son autonomie propre, mais prend sa signification seulement par rapport à la fonction computationnelle.

Le niveau algorithmique dépend du niveau de l'implémentation, car les opérations algorithmiques possibles dépendent du niveau implémentatif qui impose une série de contraintes opérationnelles. Mais un algorithme peut être implémenté dans de nombreux ordinateurs, de nombreux câblages neuro-naux, autant qu'il est possible de construire d'architectures matérielles pouvant exécuter cet algorithme. Le niveau de l'implémentation ne détermine donc pas le niveau algorithmique.

Le niveau implémentatif dépend du niveau algorithmique et du niveau computationnel pour pouvoir définir les opérations et les fonctions mentales.

Le niveau computationnel dépend du niveau algorithmique pour mettre en oeuvre les opérations de traitement d'information nécessaires. Le niveau algorithmique donne également l'ensemble des possibilités de traitement.

Y a-t-il une dépendance entre le niveau de l'implémentation et le niveau computationnel ? Le fonctionnalisme répond par la négative, et propose une indépendance du niveau computationnel par rapport au niveau de l'implémentation [25, p.110]. Le niveau de l'implémentation doit-il être négligé au profit du seul niveau computationnel ? Le niveau de l'implémentation impose des contraintes en ce qui concerne la façon d'implémenter le niveau computationnel, mais est-ce son seul rôle ? Le niveau computationnel, autrement dit, est-il un domaine indépendant des deux autres ? Non : l'étude de l'implémentation donne des indications sur la fonction à réaliser, parce qu'un critère indépendant d'optimalité n'est pas toujours vérifié par l'évolution naturelle. En particulier, le critère de l'optimum n'est pas nécessairement le meilleur critère. L'évolution biologique s'est souvent adaptée aux nouveaux problèmes en utilisant les ressources qui étaient actuellement disponibles, et les capacités cognitives sont le fruit d'une longue évolution ; les stratégies développées doivent donc être prises en compte par le niveau computationnel, au risque d'utiliser des critères (comme celui de l'optimum) qui n'est pas en jeu dans les systèmes cognitifs.

Entre les niveaux fonctionnel, implémentatif et algorithmique, il existe une forte interdépendance, qui laisse entrevoir une coopération intense entre l'intelligence artificielle, les neurosciences, et les sciences cognitives.

Nous avons vu dans cette section que les sciences cognitives conçoivent un niveau cognitif spécifique au mental vu comme pleinement autonome par rapport aux processus neuronaux ou physiques, par le biais du concept de compu-

tation. En essayant d'adopter des approches purement implémentationnelles ou purement algorithmiques, un troisième niveau computationnel se dessine, et émerge alors le concept de computation. Ce niveau est autonome, mais pas indépendant : chacun des trois niveaux que nous avons décrit dépend des deux autres.

Spécificité de la modélisation biologique

Cette réflexion sur la nature des sciences cognitives nous permet de mettre à jour une méthode qui est à l'oeuvre dans le cadre de l'agence anthropique. Il semble en effet qu'il faille deux étapes principales lorsque l'on modélise :

1. La détermination précise de la fonction que l'on souhaite reproduire.
2. La réalisation effective de cette fonction avec les moyens disponibles aussi bien au niveau algorithmique que d'un point de vue matériel.

Lassègue [17, p.21] exprime ce même fait en d'autres termes : «On doit donc considérer la démarche de la modélisation en intelligence artificielle comme incluant deux étapes bien distinctes : une étape abstraite au cours de laquelle on doit poser un problème par rapport à une fonction déterminée ; une étape plus technique dans laquelle on doit se demander *comment* peut s'incarner dans un mécanisme de type algorithmique la fonction décrite en termes calculables.».

La méthodologie qui entoure l'agence anthropique est la même. D'abord l'étape abstraite qui isole la fonction à réaliser. Quelle est cette fonction ? Il s'agit de la courbe de glucose chez une personne saine. Ainsi l'on doit réaliser un système qui réalise la fonction du pancréas, plus précisément le fonction de rendre la variation de glucose dans le temps conforme à la courbe standard. Il s'agit bien là du niveau computationnel, puisque nous avons posé une fonction à réaliser. Ensuite l'étape de réalisation effective. Dans notre cas, cette réalisation effective est contrainte par le fait d'utiliser impérativement le concept d'agence.

6.4.2 L'expérience de pensée

Une des caractéristiques essentielle de la scientificité moderne est la notion d'*expérience de pensée*.

Lassègue remarque combien la notion d'hypothèse dans l'antiquité diffère de sa compréhension moderne. La réalité physique peut être décrite, par exemple, en termes géométriques, qui étudient les rapports entre des objets idéaux. Mais les anciens ne croient pas en la vérité de ces hypothèses

mathématiques ou géométriques, car ils considèrent que le réel est essentiellement irrationnel, et que cette mathématisation de la réalité est un moyen de «sauver les phénomènes». Autrement dit, la valeur de vérité d'une hypothèse ne peut être connue.

Dans l'attitude moderne scientifique, au contraire, on suppose d'emblée que les hypothèses sont vraies, et l'on tente de le prouver par des montages expérimentaux. La nature sert alors d'extériorité qui contrôle la validité de l'hypothèse. Galilée est le véritable initiateur de ce nouveau monde de pensée. La différence avec l'attitude antique est cette volonté de réintégrer les hypothèses émises dans le cadre du vrai et du faux, c'est-à-dire de leur donner une valeur de vérité. Le sujet reconstruit la réalité physique à partir d'hypothèses, et tente ensuite de trouver confirmation de ses hypothèses dans la nature.

Nous voyons donc que l'accent est mis sur la construction *a priori*, c'est là ce que l'on appelle l'*expérience de pensée*. On effectue déjà a priori des expériences, sans aucune référence à la réalité physique concrète.

Mais cette intériorisation de l'expérience est également manifeste dans la conception de la mathématisation. L'antiquité ne faisait qu'appliquer les outils mathématiques à la réalité telle qu'elle est perçue. Lassègue nous propose ainsi l'exemple de Kepler qui applique la théorie des polyèdres réguliers pour expliquer l'agencement des planètes. L'attitude moderne est tout autre, et une de ses premières caractéristiques est justement la méfiance à l'égard du perçu. On transforme au préalable la réalité en rapports mathématiques d'objets idéalisés.

La générativité manifeste encore cette différence de la mathématisation entre antiquité et modernité. La générativité, c'est la capacité d'une théorie à produire des conséquences, et par là des prévisions. La mathématique antique est sans générativité, parce que la description mathématique du perçu ne le permet pas ; la mathématisation moderne permet une générativité car elle est une re-construction de la réalité, reconstruction qui se situe, comme nous l'avons vu dans la section 6.3, dans un champ d'opérations intellectuelles.

Aussi bien dans le rapport à la notion hypothèse et à la mathématisation, la science moderne met l'accent sur l'expérience de pensée. Le champ d'opérations intellectuelles des sciences empiriques est très exactement cette expérience de pensée, qui consiste à reconstruire le réel et à utiliser la générativité de cette construction pour la manipuler et en obtenir des conséquences. Ce qui infirme ou confirme cette construction, c'est bien sûr la réalité qui, au travers du modèle, sert de contrôle de la validité des hypothèses.

6.4.3 Nature de l'ordinateur

Avant de préciser la thèse que nous voudrions défendre concernant la modification du schéma des sciences empiriques, il nous semble important de préciser la nature de l'ordinateur, et son originalité par rapport à l'ensemble des outils créés jusqu'ici par l'homme.

Ce qui fait l'originalité de cet outil, c'est qu'il réifie la pensée (disons fabricative) sans avoir affaire à autre chose que cette pensée. Dès que nous avons un modèle mathématisé, sous forme de système formel, nous sommes capable, au travers de la machine universelle de Turing, de rendre extérieur ce système formel dans une machine, c'est-à-dire de le réifier. Lorsque cette machine est un ordinateur, on est alors confronté seulement avec les pensées fabricatives elles-mêmes, et nous ne sommes confrontés à aucune extériorité physique. Pour préciser cette nature, prenons l'exemple d'un modèle d'un briquet. Supposons que nous ayons pu en faire un modèle formel très précis. Nous pouvons alors le réifier. La première possibilité est de reconstruire un briquet réel à partir de ce modèle formel. C'est là également une extériorisation de la pensée humaine, parce que l'objet construit le sera par la pensée humaine. La seconde possibilité est la réification de ce modèle formel sous la forme d'un programme dans une machine universelle de Turing. Dans les deux cas nous retrouvons une extériorisation et une réification de la pensée, mais l'on voit bien que ces deux cas diffèrent. Dans le premier cas, il faut utiliser le donné de la nature pour arriver à ses fins, il faut être soumis autrement dit aux lois de la nature, au donné en général que nous propose le monde. Dans le second cas, nous avons seulement affaire au modèle formel lui-même, aux symboles et aux concepts qui y sont attachés, nous extériorisons ce système sans rien d'autre, nous ne sommes pas soumis aux lois de la nature et nous n'avons rien affaire avec le donné du monde.

Le briquet réifié dans une machine universelle de Turing n'est d'ailleurs pas un briquet : physiquement, il n'y a qu'une concrétisation particulière du concept de machine universelle de Turing chargée avec un programme.

On peut essayer d'objecter qu'il y a bien une soumission aux lois de la nature par le biais de l'ordinateur physique qui fait fonctionner le programme. L'argument serait qu'il existe bel et bien une partie physique contraignante qui est celle de l'ordinateur. Mais ce serait oublier la genèse de cet outil concret que nous appelons ordinateur : l'ordinateur est essentiellement un concept, à savoir la machine universelle de Turing, qui n'est pas une machine concrète mais qui fixe un certain nombre de principes. Ces principes, s'ils sont repris dans une machine concrète respectant les principes théoriques de la machine universelle de Turing, est alors une machine universelle de Turing (bien que l'ordinateur souffrira toujours d'une contrainte supplémentaire

par rapport à la machine universelle de Turing, qui possède des tables d'instruction infinie). Weizenbaum [26] faisait déjà remarquer qu'un rouleau de papier découpé en carré avec des cailloux ingénieusement agencés constituait une concrétisation d'une machine universelle de Turing.

Des exemples de cette indépendance physique ne manquent pas. Prenons celui de la loi de gravitation universelle qui dépendrait non pas du double de la distance, mais bien du triple. La réification d'un tel modèle est possible dans un ordinateur, mais elle est évidemment inconcevable dans l'extériorité de la nature.

On voit d'ailleurs dans cet exemple combien le terme *réification* est ambigu dans le cas de la machine universelle de Turing : on ne réifie pas à proprement parler une chose, puisqu'il paraît totalement impossible de concrétiser une chose qui ne respecterait pas les lois de la nature. En fait, la seule chose que l'on concrétise c'est un exemple de la machine universelle de Turing, et ce sont ses états que nous faisons correspondre au modèle que nous en avons.

Si l'ordinateur est bien une machine concrète, soumise aux lois de ce monde, le programme quant à lui est complètement indépendant de telle ou telle machine concrète, et peut donc fonctionner sur chacune d'entre elles. C'est la raison pour laquelle le programme n'a pas à faire face à des contraintes physiques : il est seulement un système formel jouant avec ses propres symboles.

Nous traduisons cette originalité de l'ordinateur en disant que tout programme est une *abstraction*.

6.4.4 Extériorité interne de la pensée

Dans le schéma des sciences empiriques, l'accent sur l'expérience de pensée est accompagnée d'une extériorité de contrôle et cette extériorité de contrôle est la réalité. La thèse de Lassègue, que nous voudrions utiliser et développer, est qu'avec l'ordinateur, l'extériorité n'est plus celle de la nature mais est une extériorité interne de la pensée.

La première étape de l'activité des sciences empiriques consiste dans la recherche d'hypothèses et dans l'élaboration d'une théorie qui aboutit généralement à la constitution d'un modèle, comme nous l'avons vu dans la section 6.3. Une fois ce modèle formalisé, il est extériorisé dans un ordinateur. C'est l'ordinateur au terme du fonctionnement du modèle qui fournit des résultats sur base desquels on pourra modifier les hypothèses et la théorie.

De là une conséquence importante : c'est le programme fonctionnant sur l'ordinateur qui devient l'objet duquel on attend une réponse, et avec lequel on modifie la théorie. En somme, c'est le programme lui-même qui devient l'extériorité qui contrôle la validité ou la fausseté des hypothèses ; dans ce

cas, l'extériorité n'est plus la réalité, mais la pensée elle-même, la pensée formelle réifiée dans l'ordinateur. On laisse en fait aller le programme à sa générativité, pour voir ce qu'il va donner, et l'on compare avec ce que l'on attendait. L'extériorité n'est plus celle de la nature mais est une extériorité interne de la pensée.

L'IA relie entrée et sortie au moyen d'un système formel qu'elle extériorise sur dans une machine universelle de Turing. Le programme étant abstraction, elle a seulement affaire à sa pensée. Il n'y a pas d'objet à manipuler, il n'y a pas de lois de la nature à manipuler : il y a seulement la pensée au prise avec la pensée formelle (ou une partie de la pensée formelle, puisque certains programmes sont NP-complets). L'objet devient la pensée elle-même, réifiée dans un objet capable de réitérer sans cesse cette pensée.

C'est pourquoi la notion d'expérience change radicalement. Ce n'est plus la nature qui nous donne une réponse, mais c'est notre pensée fabricative elle-même qui devient l'objet du test. L'observé est l'observateur : dans la machine de Turing chargée du programme, nous testons les conséquences de notre propre pensée.

La modification du schéma des sciences empiriques consiste donc en la mise entre parenthèses de l'extériorité de la nature au profit de l'extériorité interne de la pensée.

6.4.5 Intelligence Artificielle et sciences empiriques

L'IA se veut une science expérimentale, souhaitant bénéficier de toute la certitude attachée à la méthodologie expérimentale des sciences cognitives. On retrouve cette affirmation dans une caractérisation de la discipline [24, p.1] : «Observons que l'IA est une discipline expérimentale, dans le sens qu'un système réalise les objectifs demandés seulement lorsque les prestations désirées sont effectivement expérimentables». Cette remarque s'applique d'ailleurs à l'ensemble des articles décrivant un système, puisque l'on y trouve systématiquement une partie nommée "résultats expérimentaux". Si l'intelligence artificielle se base sur la méthode expérimentale, doit-on la considérer comme une science expérimentale ? Pour justifier sa pleine appartenance au paradigme scientifique courant, l'intelligence artificielle insiste sur le caractère réel et concret de la machine universelle de Turing chargée d'un programme ; elle insiste aussi sur le fait qu'une fois réifié, elle devient un objet d'expérience comme tout autre objet du monde réel. L'objet de l'intelligence artificielle est un objet du monde, et la certitude que l'on acquiert est identique à la certitude que l'on acquiert en testant en sciences empiriques. Si la machine réifiée donne la fonction que l'on voulait lui assigner, alors l'objectif est atteint : on teste l'objet et il agit comme on l'avait prédit. Notre théorie

est donc exacte.

Mais en voulant s'identifier aux sciences empiriques, l'intelligence artificielle nie la spécificité de son objet. Cet objet est en fait la pensée fabricative réifiée, il est le simulacre de la pensée, et son objet réifié a donc un statut différent de l'objet des sciences empiriques. Le principe d'une science empirique, c'est d'avoir la réalité comme objet, le monde avec ses lois, c'est faire face à cette extériorité et lui demander des comptes. L'intelligence artificielle a la pensée fabricative comme objet, et c'est une extériorité particulière : à vrai dire, il ne s'agit pas d'une extériorité, puisque il s'agit de la pensée elle-même (ou une partie de celle-ci). Autrement dit, ce n'est pas un objet extérieur, une transcendance que nous testons, mais *nous-mêmes*, c'est-à-dire la pensée fabricative.

Si l'on veut affirmer que l'intelligence artificielle est une science empirique, il semble qu'il suffit qu'elle emploie la même méthode. Alors l'intelligence artificielle se force à plier sa méthode à celle des sciences empiriques, en faisant de la pensée réifiée un objet du monde réel, comme s'il nous était radicalement extérieur, comme s'il n'avait pas de lien avec nous, de sorte que cette pensée réifiée obtienne le même statut que l'objet des sciences empiriques. Après quoi il devient évident que l'intelligence artificielle est effectivement une sciences empirique, elle expérimente un objet extérieur, transcendant du monde réel.

Seulement voilà : cette transcendance si évidente dans le cas des sciences empiriques entre le sujet et l'objet n'est plus de mise lorsque l'on parle de l'intelligence artificielle et de la machine universelle de Turing. Le rapport sujet/objet est transformé.

Remarquons d'abord que cette transcendance dans les sciences empiriques est elle-même problématique. C'est un lieu commun aujourd'hui d'affirmer qu'en science le sujet et l'objet sont liés ou que le sujet détermine en partie l'objet. L'exemple classique est celui de l'observation des atomes : le fait de les observer modifie la disposition des atomes, et l'on ne sait plus alors si l'objet se donne au sujet ou si le sujet détermine l'objet. C'est une question épistémologique que nous n'abordons pas en tant que telle ici. Notre but est simplement, dans ce paragraphe, de remarquer que ce lien entre sujet et objet est déjà présent dans les sciences empiriques.

Avec l'intelligence artificielle, cette transcendance (relative) disparaît : l'objet fait ce que le sujet fait. L'objet est la réification de la pensée du sujet. Tester cet objet est tester la pensée du sujet lui-même : d'une extériorité de la nature (on teste la nature) on passe à une extériorité interne à la pensée (on teste la pensée elle-même). L'objet est alors immanent au sujet : l'objet est le sujet, ou plutôt : l'objet incarne des opérations symboliques qui sont des représentations symboliques de la pensée du sujet.

La conséquence de ces considérations sur le rapport entre sujet et objet est qu'avec l'intelligence artificielle il n'y a plus de donné, d'extériorité transcendante comme dans le cas des sciences empiriques. Il n'y a plus la manipulation de quelque chose d'extérieur à la pensée humaine. Il s'agit de maîtriser notre pensée formelle pour atteindre un certain objectif, en l'extériorisant dans une machine universelle de Turing.

6.4.6 Le double rôle de la machine universelle de Turing

Si la machine universelle de Turing est bel et bien la réification de la pensée fabricative, alors nous voyons que la machine universelle de Turing possède deux rôles. D'abord elle réalise la fonction initialement demandée. Ensuite elle a besoin d'un système formel reliant entrée et sortie, elle est donc un modèle de cette fonctionnalité. En effet rappelons qu'un modèle est le point de vue du concepteur omniscient qui permettrait au sujet de reconstruire l'objet. Mais dans notre cas l'objet est la production du sujet, c'est le sujet qui réifie sa pensée pour donner lieu à l'objet, et il en possède donc automatiquement le modèle.

6.5 Application de ces concepts

6.5.1 Le cas concret de l'agence anthropique

Le processus métabolique du glucose

Le cas applicatif du concept de l'agence et de l'agence anthropique est celui du processus métabolique du glucose. L'objectif de cette modélisation est double :

1. Utiliser le concept d'agence, tel que décrit dans le chapitre 2.
2. Modéliser le processus biologique du métabolisme du glucose, processus généralement mené par le pancréas. L'intérêt d'avoir choisi ce processus est qu'il recouvre en fait le problème du diabète.

Rappelons que les modèles mathématiques des processus physiologiques passent d'abord par un générateur qui produit des modèles utilisables par les agents décisionnels (voir chapitre 3).

La figure 6.1 résume ainsi le processus global de modélisation et de réification. Les modèles M_1, \dots, M_n se situent au carrefour de la réalité et de la théorie. Ces modèles M_1, \dots, M_n sont ensuite transformés par le générateur, et chaque modèle résultant M'_i est incorporé dans un agent décisionnel A_i que

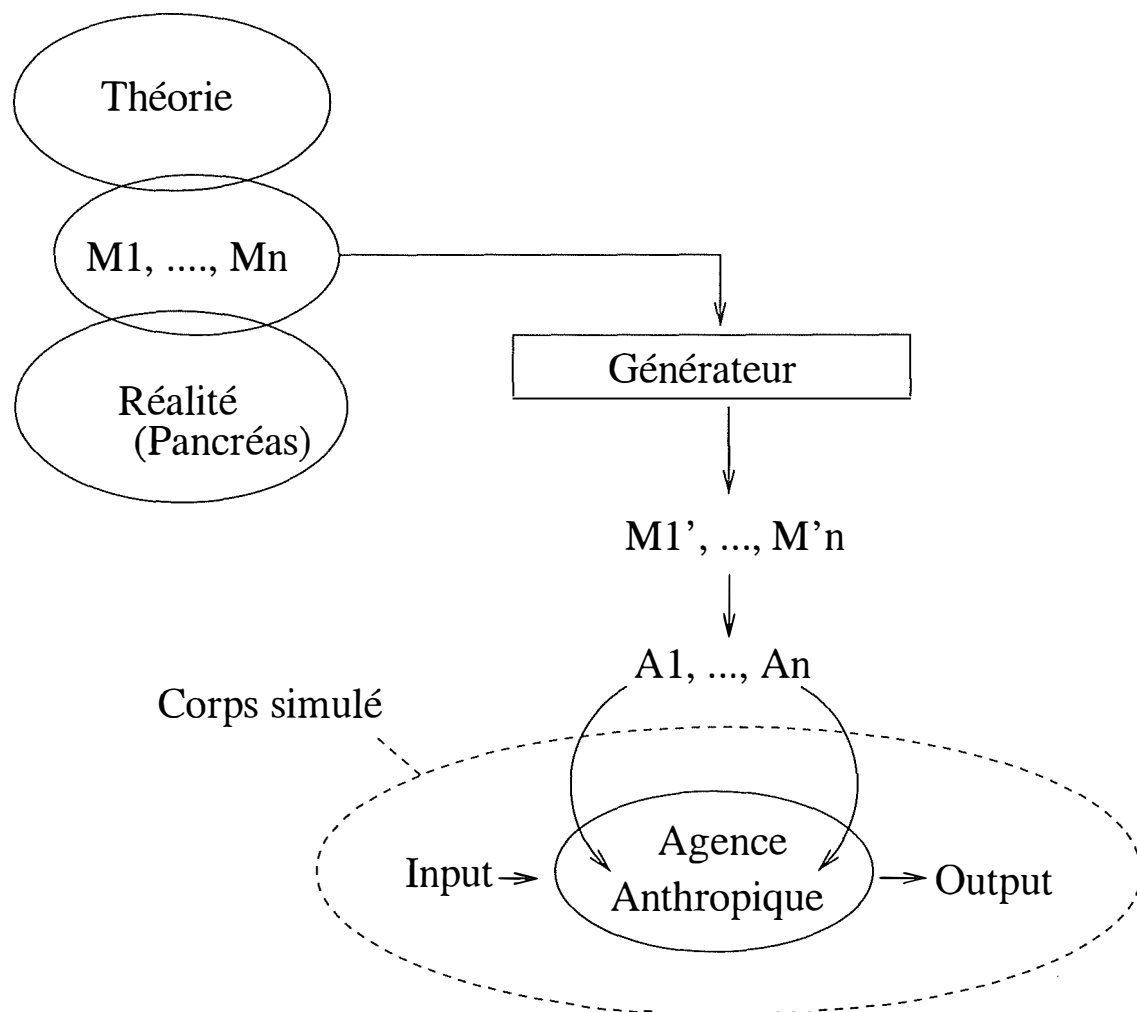


FIG. 6.1 – Schéma global de modélisation et de réification.

l'on place dans l'agence anthropique. Remarquons que le corps est seulement simulé, fournissant le milieu d'opération de l'agence anthropique.

Description des agents

L'agence anthropique est une agence telle que décrite dans le chapitre 2.

Elle possède un agent extracteur qui fournit quatre paramètres aux agents décisionnels :

1. Le niveau courant d'insuline
2. Le niveau courant de glucose
3. La variation de glucose par rapport à l'échantillonnage précédent

4. Le niveau courant d'activité physique.

Il existe également deux modèles embarqués dans des agents décisionnels.

1. Le premier renvoie une valeur de bien-être en fonction du taux de glucose, du taux d'insuline, et de la variation de glucose par rapport à l'échantillonnage précédent.
2. Le second renvoie une valeur de bien-être en fonction du taux de glucose, du taux d'insuline, et du taux d'activité physique.

Après l'arbitrage entre les différents agents décisionnels, les agents actuateurs réalisent enfin la décision de l'agence anthropique. Dans notre cas particulier, il existe un seul agent actuateur qui injecte de l'insuline.

6.5.2 La contrainte du discret dans la représentation

Rappelons la thèse du chapitre sur la représentation : l'utilisation de l'ordinateur implique une structuration de toute représentation en trois registres : le formel, l'intuitif, et le physique, auxquels correspondent respectivement le discret, l'infini et le continu. Les notions d'intuition et de physique sont des dépassements du cadre fondamentalement discret du paradigme de l'ordinateur.

Nous voudrions illustrer cette thèse dans le cas de l'agence anthropique, et souligner en particulier le lien de dépassement entre l'ordre du discret (formel) et l'ordre du continu (physique).

En effet, l'une des préoccupations majeures dans le générateur de modèle est justement la discrétisation. Les modèles produits fournissent le contenu que les agents décisionnels pourront utiliser pour communiquer une décision. Mais les modèles mathématiques d'une partie du processus métabolique ont gardé leur caractère continu, c'est-à-dire que les variables appartiennent à l'ordre des réels. Dit autrement, les modèles ne sont pas encore sous la contrainte du paradigme de l'ordinateur. Le modèle a en partie à faire avec la réalité, à travers l'expérience, et garde de celle-ci un caractère continu. Par exemple, la variable décrivant le taux de glucose est continue, et l'on peut découper ce glucose en une infinité d'intervalles.

Rappelons que les modèles produits ont des dimensions qui correspondent aux variables importantes du processus métabolique considéré. A chaque point de cet espace multi-dimensionnel correspond une valeur de bien-être (voir section 3.4). Mais ces dimensions ont des valeurs continues, et elles peuvent être infiniment divisées. Il faut *nécessairement* les discrétiser, parce qu'aucune opération discrète ne peut venir à bout d'un intervalle continu. La remarque est explicitement dite dans l'article sur le générateur de modèles :

«The implementation problem is due to the discretization of the variables; the discretization of the model is necessary to generate a finite and utilisable representation.» [13, p. 6].

Cette restriction au discret ne doit pas être prise à la légère. Elle génère en effet des problèmes qui n'existent pas en dehors du paradigme de l'ordinateur. Dans l'exemple du générateur de modèles, la discrétisation des dimensions (glucose, insuline, etc.) implique des conflits entre les points solutions qui tombent dans les mêmes cases de l'espace-multidimensionnel alors que ce sont, du point de vue du modèle mathématique encore dans le continu, des points différents (voir section 3.4.4, p. 63). Plusieurs points de degré de pathologie différent peuvent également se retrouver au même endroit. Il faut alors se donner une règle pour choisir un de ces points; dans ce générateur ce critère est paramétrable, mais nous avons choisi de garder le point avec le plus bas degré de pathologie. Ce qu'il faut souligner, c'est que cette règle n'existe pas dans le modèle initial : il s'agit d'une déformation due à l'utilisation du paradigme de l'ordinateur.

Cette discrétisation des intervalles pose encore d'autres problèmes. Puisque le modèle multi-dimensionnel est «quadrillé», beaucoup de points vont se retrouver dans la même case, et des cases risquent de se retrouver vides à cause de cette discrétisation arbitraire. Mais surtout, cette discrétisation ne convient pas aux agents décisionnels eux-mêmes : ceux-ci font en effet une recherche descendante dans cet espace et le fait que certains points ne soient pas modifiés empêche toute recherche descendante, car ces points valent la valeur maximale de pathologie (100), et certains points de degré de pathologie minimal (1) risquent d'être entourés de tels points.

On décide alors d'introduire le concept de déflagration qui trace un dégradé autour des points insérés de sorte que le résultat soit utilisable par les agents décisionnels. Cette déflagration pose l'hypothèse, réaliste au demeurant, qu'autour des points de solution le degré de pathologie est voisin et diminue au fur et à mesure que l'on s'éloigne de ce point.

Que penser d'une telle hypothèse? Remarquons tout d'abord que cette hypothèse se base non pas sur le modèle initial, mais sur la représentation de ce modèle en terme de machine. Même si elle semble raisonnable, aucune donnée scientifique n'a été utilisée pour la prouver.

Mais l'idée de déflagration engendre encore d'autres problèmes : que se passe-t-il en effet lorsque deux déflagrations se recouvrent? Voilà un cas qui non seulement n'a aucun lien avec le modèle initial mais qui de plus n'existe pas dans la réalité objective qu'étudient ces modèles. Que faire en effet d'un tel conflit qui n'existe que dans la contrainte du discret? La résolution de ce conflit doit nécessairement être arbitraire ou subjective, puisque cela n'existe pas dans la réalité objective. Ce critère de résolution est paramétrable dans

le générateur et nous avons choisi de toujours garder le degré de pathologie le plus bas.

Une critique pourrait nous être adressée : le générateur de modèles n'est pas nécessaire, on aurait pu donner le modèle mathématique directement aux agents décisionnels qui auraient déterminé directement le degré de pathologie par calcul. L'utilisation d'une discrétisation n'est pas donc nécessaire, et les problèmes rencontrés dans le générateur sont évitables. Une telle objection manque en fait l'essentiel. D'abord il faut souligner qu'une utilisation directe de ce modèle mathématique ne fait pas partie des méthodes canoniques de l'intelligence artificielle : recherche dans un espace de points, transition d'états, etc. Mais surtout l'utilisation du modèle mathématique original n'efface pas la contrainte du discret : les courbes d'entrée du modèle devraient toujours être discrétisées, les paramètres devront toujours être discrétisés, et les solutions seront encore de nature discrète.

Que conclure de cette analyse du générateur ? Deux choses. D'abord l'évidente discrétisation du modèle initial. On voit dans notre exemple un cas concret de l'influence du cadre du discret au sein du paradigme de l'ordinateur. Mais ce cas concret montre aussi le changement du rapport entre sujet et objet. L'objet fait ce que le sujet fait. Comme nous extériorisons la pensée en vue d'atteindre un objectif, comme nous ne faisons plus face au donné du réel, il est alors possible d'introduire des éléments qui n'appartiennent pas au modèle initial et qui n'appartiennent qu'à la pensée elle-même, par exemple le concept de déflagration ou le critère pour régler les conflits, etc. Tous ces concepts et mécanismes n'existent que pour le sujet lui-même qui décide des critères à appliquer, parce que ces critères n'ont aucune réalité objective. Ces problèmes qui ne se posent que pour le sujet et qui sont réglés par des mécanismes de pensée du sujet seul sont la trace de cette disparition de la transcendance dont nous parlions à la fin de la section 6.4.5. Le sujet a seulement affaire à sa propre pensée dans cet objet qu'est l'ordinateur.

6.5.3 Le double rôle de l'Agence Anthropique

Nous voudrions à présent critiquer le double rôle assigné à l'agence anthropique : le rôle de description et le rôle de soutien à la découverte scientifique. Reprenons ce double rôle à la lumière de notre thèse, qui voit l'ordinateur comme l'extériorisation de la pensée du sujet. Nous avons déjà souligné une conséquence importante de cette thèse : la machine universelle de Turing implique nécessairement la création d'un modèle, lorsque l'on construit un programme pour réaliser un objectif fixé.

On voit que le double rôle de l'agence, affirmé comme une originalité du concept d'agence (voir section 4.2.3), est en fait inscrit dans la nature même

de la machine universelle de Turing. La machine universelle de Turing possède ces deux rôles : elle fournit un modèle et elle vise un objectif. Autrement dit, dès que l'on écrit un programme pour un objectif fixé, on obtient un modèle. Ce que l'on prend pour une caractéristique propre à l'agence est en fait seulement une caractéristique inscrite dans la machine universelle de Turing.

Lorsque l'on se donne la coopération entre les modèles comme moyen d'implémentation pour réaliser un objectif (ce qui est l'idée de l'agence) on obtient de toute nécessité un modèle, qui par ailleurs peut fort bien ne correspondre à rien de réel, et n'être qu'une liaison quelconque de jeux symboliques entre entrées et sorties.

C'est pourquoi, si le rôle de soutien à la découverte scientifique semble évident, surtout avec l'usage que l'on fait aujourd'hui de l'ordinateur en science, le second rôle de description est loin d'être acquis. Si nous obtenons automatiquement un modèle de la coopération de modèles, il reste à savoir si ce modèle que nous obtenons est un modèle adéquat. Créer un jeu de symboles, même sous la forme d'une coopération de modèles, n'assure pas l'adéquation du modèle et de son objet.

Le cas de l'agence anthropique est assez clair à cet égard. Nous devons demander : le fonctionnement de l'agence anthropique constitue-t-il une description opérationnelle du résultat de l'activité scientifique ? La question est ambiguë. Si n'importe quelle description opérationnelle convient, alors la réponse est certainement oui. On peut supposer que toute description opérationnelle ne convient pas, car sinon n'importe quel jeu de symboles respectant la coopération des modèles est une description adéquate. Il faut donc se demander ce que l'on peut considérer comme une description opérationnelle adéquate. Est-ce qu'un ensemble d'agents logiciels, recherchant le plus bas degré de pathologie dans un espace multi-dimensionnel et se mettant d'accord sur une décision via un mécanisme complexe de coopération tel que décrit dans le chapitre 2 constitue une description opérationnelle adéquate ? Cette question importante reste en suspens, car c'est aux scientifiques eux-mêmes d'y répondre.

L'équipe de Milan traiterai cette critique comme négligeable, car le mécanisme de coopération ainsi que les agents décisionnels eux-mêmes pourraient toujours être programmés pour respecter un certain type de description choisi à l'avance. Radicalisons dès lors la critique.

Deux remarques s'imposent. D'abord le caractère opérationnel de la description est lui-même problématique, et ce caractère opérationnel est essentiel à la machine universelle de Turing ; il est donc impossible à l'agence de s'en défaire. Il faudrait se demander : une description opérationnelle est-elle un type de description acceptable pour le résultat de la découverte scientifique ? Jusqu'ici l'humanité a semblé considérer que le résultat de la découverte

scientifique devrait être représenté par des textes et des métaphores. Le mode de description de l'agence en somme semble être un nouveau mode jamais utilisé. Il reste à savoir si ce mode lui-même est un mode adéquat pour la description du résultat de la découverte scientifique.

Une seconde remarque, plus importante, éclairera ce qui vient d'être dit. Le mode opérationnel de la description est de nouveau déterminé par la nature de la machine universelle de Turing. Elle est, en effet, de nature opérationnelle : elle effectue des opérations symboliques. Si elle implique automatiquement la création d'un modèle, elle implique aussi nécessairement un caractère opérationnel à ce modèle.

On peut encore généraliser la remarque précédente : le rôle de description de l'agence est secondaire par rapport au problème technique d'implémentation. On essaye *d'abord* de réaliser un programme, sous la contrainte de la coopération de modèles, on adapte les données, les algorithmes, etc. à l'obtention de l'objectif, et on obtient après coup une description du résultat de la découverte scientifique. On décide, par exemple, d'utiliser une recherche descendante dans les agents décisionnels, sans aucune préoccupation de la pertinence descriptive ; on adapte les modèles à cette recherche descendante au travers d'un générateur, etc. On met en avant ensuite le rôle descriptif de cette agence. Et cela, nous le répétons, à cause de la nature de l'outil utilisé, la machine universelle de Turing qui produit de fait un modèle opérationnel.

Il faudrait renverser cette approche, et d'abord poser la question d'une description opérationnelle adéquate du résultat de la découverte scientifique, *si* un tel mode opérationnel a une pertinence quelconque pour le savoir humain. Rien n'assure, par ailleurs, qu'une telle description adéquate pourrait être implémentée ; il se pourrait par exemple que l'algorithme de cette description soit NP-complet.

On pourrait faire des remarques semblables au sujet de l'agence représentationnelle. La richesse d'un modèle est décrite selon le nombre de symboles du modèle, mis sous forme canonique. De nouveau ce critère est utilisé parce qu'il est mesurable par une machine universelle de Turing, puisqu'elle manipule elle-même des symboles, à l'instar des systèmes formels. Un tel critère de richesse des modèles est discutable, et un critère de richesse plus communément admis est celui du pouvoir prédictif du modèle ; mais un tel critère n'est pas utilisable par la machine universelle de Turing.

6.5.4 Signification épistémologique de l'agence anthropique

Il nous reste à montrer en quoi la modélisation biologique, lorsqu'elle veut remplacer un organe par une machine utilisant une machine universelle de Turing, illustre la thèse selon laquelle en intelligence artificielle l'objet n'est plus transcendant au sujet.

Dans l'agence anthropique, cette transcendance entre le sujet et l'objet disparaît dès que les agents extracteurs ont numérisé des informations sur le corps. A partir de ce moment, l'information numérisée sur le monde est soumise uniquement à l'ordre de la pensée, c'est-à-dire que ces informations sont transformées en symboles manipulables par notre pensée réifiée. Le pancréas réel est remplacé par une machine universelle de Turing, par une agence, et nous pouvons alors manipuler avec notre propre système formel ces informations, sans plus être soumis aux lois de la nature ni aux contraintes physiques. Cette manipulation, de plus, se fait dans le cadre du paradigme de l'ordinateur, avec sa contrainte du discret.

Une fois la manipulation effectuée, l'information est alors rendue de nouveau réelle par l'intermédiaire des agents actuateurs.

La signification épistémologique de ce remplacement d'un organe par une machine universelle de Turing est que l'homme écarte la transcendance du monde en remplaçant le réel par la pensée fabricatrice de l'homme.

De cette manière, l'objet devient le sujet. Avec l'ordinateur, nous sommes devenus capables de créer nos propres modèles du réel, nous sommes devenus le créateur omniscient dont parlait Ladrière.

Conclusion

Au terme de ce parcours à travers le paradigme de la coopération en intelligence artificielle, de la description de l'agence anthropique, et de nos réflexions épistémologiques, nous sommes en mesure de répondre à notre question de départ : en quoi la méthode de l'intelligence artificielle se différencie-t-elle de la méthode des sciences expérimentales ?

La réponse principale que nous avons découverte est celle de la disparition de la transcendance (voir section 6.4.5, page 107) de l'objet par rapport au sujet en intelligence artificielle, contrairement aux sciences empiriques où l'objet reste dans une transcendance relative par rapport au sujet.

A cet égard, il nous semble important de souligner le rôle de la simulation. Lorsque l'on dit que l'on *simule* un modèle sur un ordinateur, on signifie par là que l'on «fait semblant», c'est-à-dire que l'on joue avec soi-même, plutôt que de se laisser affecter par un objet externe à soi. La simulation, en intelligence artificielle comme en informatique en général, est l'acte par lequel l'homme réifie un objet dont on a retiré la transcendance initiale contenue dans l'extériorité de la nature.

Par ailleurs c'est à partir de cette disparition de la transcendance que l'on peut comprendre non seulement la méthodologie de l'intelligence artificielle mais également le paradigme de la coopération.

D'abord la réification de la pensée fabricative rend possible le paradigme de la coopération des modèles, d'une "société de l'esprit" (expression utilisée par Minsky) car nous pouvons surtout penser cette société de l'esprit ; elle serait très difficile à construire physiquement : nous aurions en effet alors à faire avec la transcendance du monde réel. Avec la machine universelle de Turing, nous avons affaire à notre propre pensée, à nos propres lois.

Ensuite ce changement de rapport entre sujet et objet fait de l'homme le concepteur omniscient du modèle. De là le double rôle de la machine universelle de Turing et le double rôle que l'on assigne à l'agence. Autrement dit, le double rôle de l'agence qui semble être une *découverte* est déjà inscrit dans la nature même de la machine universelle de Turing : sa capacité à réifier la pensée.

La seconde réponse que nous avons découverte en ce qui concerne la méthodologie de l'intelligence artificielle est le caractère discret du paradigme de l'ordinateur, qui implique une déformation des modèles initiaux.

Enfin nous avons insisté sur le caractère opérationnel de la machine universelle de Turing qui rend la description assumée par l'agence de fait opérationnelle.

Ce que nous voyons, c'est que la nature de la machine universelle de Turing détermine la méthode et le contenu de l'intelligence artificielle. La méthode : l'intelligence artificielle est immédiatement inscrite dans le cadre du discret et elle a tendance à s'éloigner des modèles réels parce qu'elle manipule sa propre pensée. Le contenu : l'intelligence artificielle a tendance à réduire tout objet extérieur en fonction de la nature de son cadre de travail, la machine universelle de Turing, comme il apparaît dans la discrétisation des modèles, dans la tentative de réduire la description du savoir scientifique à une description *opérationnelle*, à la mesure de la richesse d'un modèle scientifique et donc du nombre de symboles de ce modèle, etc.

La machine universelle de Turing est un outil original créé par l'homme qui lui ouvre un nouveau champ de possibilités qui a ses propres limites et apories. Le nouveau rapport entre sujet et objet qu'elle implique, son caractère discret et opérationnel dresse un cadre de travail a priori dans lequel les chercheurs en intelligence artificielle sont plongés à leur insu.

Terminons ce travail par une remarque sur la signification générale de l'utilisation de la machine universelle de Turing en place du monde réel. Le cas de l'agence anthropique était de remplacer un organe du corps par une machine universelle de Turing. Ce faisant on remplace un phénomène du monde par la pensée même de l'homme. Autrement dit, à travers cette machine qu'est la machine universelle de Turing, l'homme fait reculer d'un cran la transcendance du monde et la rend identique à sa pensée.

Bibliographie

- [1] Church A. A note on the entscheidungproblem. *Journal of Symbolic Logic*. vol. 3, n°1.
- [2] Turing A.M. On computable numbers with an application to the entscheidungproblem. *Proceedings of the London Mathematical Society*, (42) :230–265, 1936.
- [3] Turing A.M. Computer machinery and intelligence. *Mind*, pages 433–460, 1950.
- [4] F. Amigoni. *Dynamic Agency : a Methodology and Architecture for Multiagent Systems*. Dipartimento di Ellectronia e Informazione, 2000.
- [5] H. Bergson. *Creative Evolution*. University Press Of America, 1984.
- [6] D. Zanisi F. Amigoni, M. Somalvico. A theorical framework for the conception of agency, 1999.
- [7] M. Somalvico F. Amigoni, V. Schiaffonati. Dynamic agencies : Models for creative production and technology applications.
- [8] M. Somalvico F. Amigoni, V. Schiaffonati. Dynamic agencies and creative scientific discovery.
- [9] M. Somalvico F. Amigoni, V. Schiaffonati. A multilevel architecture of creative dynamic agency.
- [10] M. Somalvico F. Amigoni, V. Schiaffonati. A multilevel architecture of creative dynamic agency. International Congress on Discovery and Creativity. Gent, Belgium, Mai 1998.
- [11] M. Somalvico F. Amigoni, V. Schiaffonati. Dynamic agencies and creative scientific discovery. In *Proceedings of the AISB'99 Symposium on Scientific Creativity*. The Society For The Study of Artificial Intelligence and Simulation of Behaviour, 1999.
- [12] Galileo Galilei. *Dialogue Concerning The Two Chief World Systems : Ptolemaic and Copernician*. Stillman Drake, 1967.

- [13] Zampelli S. Gatti N. A generator of physiological model representations for a distributed control approach. Technical report, Politecnico di Milano, 2001.
- [14] Ladrière J. Représentation et connaissance. In *Encyclopaedia Universalis*, volume 15, pages 904–906. ?publisher?, Paris, 1985.
- [15] Ladrière J. Model, representation, and reality. In *The information society : Evolving landscapes*. Editions Springer Verlag, 1990.
- [16] Lassègue J. Note sur l’emploi et l’origine de la notion de représentation en sciences cognitives. *Intellectica*, (17) :199–212, 1993.
- [17] Lassègue J. La méthode expérimentale, la modélisation informatique et l’intelligence artificielle. *Intellectica*, (22) :21–65, 1996.
- [18] Malherbe J.-F. *Le langage théologique à l’âge de la science. Lecture de Jean Ladrière*. Ed. du Cerf, 1985.
- [19] Bunge M. *Method, model and matter*. D. Reidel Publishing Company, 1973.
- [20] M. Matarié. Behavior-based robotics. URL = <http://www-robotics.usc.edu/maja/bbs.html>.
- [21] Minsky. *La société de l’Esprit*. InterEditions, 1988.
- [22] M. Dini N. Gatti. *Agenzia antropica : un sistema a molti agenci per applicazioni di nature biomedica*. Politecnico di Milano, 2001.
- [23] Kitcher P. Marr’s computational theory of vision. *Philosophy of Science*, (55) :1–24, 1988.
- [24] M. Somalvico. *Intelligenza artificiale*, 1999.
- [25] Dedeurwaerdere T. *Action et Contexte : du tournant cognitiviste à la phénoménologie transcendante*. Olms, 2001.
- [26] J. Weizenbaum. *Puissance de l’ordinateur et raison de l’homme. Du jugement au calcul*. Ed. d’Informatique, 1981.

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année Académique 2001 - 2002
Promoteur : Professeur Jacques Berleur s.j.

Modélisation en Intelligence Artificielle
Zampelli Stéphane
Annexe - Rapport Technique

Mémoire présenté en vue de l'obtention du grade de Maître en
Informatique

A Generator of Physiological Models Representations for a Distributed Control Approach

NICOLA GATTI, STEPHANE ZAMPELLI

ARTIFICIAL INTELLIGENCE AND ROBOTICS LABORATORY, POLITECNICO DI MILANO

1 Introduction

In the context of process control, an alternative to the classical approach based on a single monolithic controller is the decentralization that conceives a phenomenon as composed of a set of sub-phenomena; in such a way a complex model that describes the process we desire to control is spread on a network of sub-models. In particular, in our approach (a multiagent system approach) we assign to each specific agent a sub-model obtained by the adopted distribution scheme. The multi agent approach for the process control offers some advantages in comparison with the classical approach: the possibility to adopt more than one model at the same time, the flexibility and integrability of the architecture are just some examples.

Each agent is an entity provided with rational capacities; thus it needs a knowledge base to accomplish its work. In this report we propose a new technique to produce a knowledge base that describes the model of a physiological process and concerns its possible variations during the time.

The main problem is to produce a knowledge base that contains information concerning a dynamic model, its variability during the time, a characterization of the variability in relation to a specific principium. Moreover we want to produce knowledge-base according to an Object-Oriented approach, in other words we try to give a set of protocols to adopt in order to produce a package that contains the representation of the model.

The first step focuses on obtaining a static representation of a dynamic process; in fact the dynamicity of the model results an obstacle to the work of the agent. The agent has to analyze the current state of the process in comparison with the model the agent embeds. A state, measured by sensors, can be viewed as a point of a specific solution (response) of the model for a particular input with and a particular set of parameters. The agent activity is to understand if the current state of the process is a state of the model, in other words if the

current state belongs to the model solutions. The problem is that, in a dynamic environment, the agent needs to calculate each admissible solution of the model in order to search the current state among them. Instead we desire a representation that does not oblige the agent to calculate each time all the admissible solutions of the model. From another point of view we can say that the agent, adopting a static representation, can analyze the current state of the process without needing any information of the previous state.

The second step focuses on describing the variability of the process. Our application context is the physiological models, in such context the variation of a process caused by the emerging of a pathology can be modeled just by the variation of some parameters. For instance, in the test cases we have used to check our system, the glucose-insulin metabolism, the model of such metabolism in a diabetic person presents the parameters of insulin secretion much lower than in the healthy one person. The variation of some parameters makes that the states in which the process can be change during the time. Thus, each different variation of the parameters generates a specific set of states allowed for the model.

The third step focuses on characterizing such states in relation to an assumed principium. The principium that interests us to adopt concerns the description of the pathology of a specific process. We need a mathematical function that expresses the relationship between the pathology degree and the entity of the variation of the parameters.

This report is organized as follows. In Section 2 we present base notions of distributed control and a summary of our approach to distributed control. In Section 3 we discuss about physiological models, and we introduce notation about that. In Section 4 we describe the model representation and the generation process that permits to obtain that. In Section 5 we report the tests about the working of the generation system. In Section 6 we report the conclusion. In Appendix A we describe the architecture of software system. In Appendix B we report the code. In Appendix

C we report the package used to visualize the model representation.

2 Distributed control

The distributed control is a novel approach to process control. It is possible to identify, in such context, two orthogonal dimensions of distribution. The former is the distribution of a complex process in a set of simpler sub-processes – usually called in literature *decentralization* [1] –, the latter is the distribution of control activities – the classical *robotic approach* for robots control [6].

2.1 Model distribution

Usually, a control system is composed of just by an unique monolithic control element, named controller, which contains the model of the entire process to control.

The base principium of distributed control is the decentralization of the control unit in a set of simpler partial controller units. Each of these simpler controller units regulates just a single part of entire system. In other words, each simpler controller is, in reality, a sub-controller of one sub-process of the entire process. A control system for the global phenomenon simply emerges from the structured interaction of the partials controller units. Hence, the idea is to divide a complex model – the model that describes the controlled process – in a set of sub-models, and to introduce a sub-model in each sub-controller.

Each function f , that represents a model, is defined in a space \mathcal{R}^n – the domain – constituted by the variables of the model. The behavior of the process simulated by the model is described by the values assumed by the function f ; such values compose a vector of \mathcal{R}^m – the codomine of f .

$$f : \mathcal{R}^n \rightarrow \mathcal{R}^m$$

The distribution criteria of f in sub-models have not definite rules yet. In a general scheme the distribution criterion is absolutely free, so each combination is consented. In this way, each distribution criterion produces a specific set of sub-models and a specific relation among them. Hence, these relations are defined by the assumed distributing scheme. The relations among the models involve the relations among the variables that define the domain and the codomine of each sub-model.

For instance, it possible that two models overlap, that a model is included completely in another, that two models are independent, etc... In Figure 1 it is presented a schematic representation of one phenomenon divided

in to four figures representing the sub-phenomena. Thus it is necessary to assume a specific principium to make a specific distribution of a model. In this way each distribution process is bind to a specific principium of distribution.

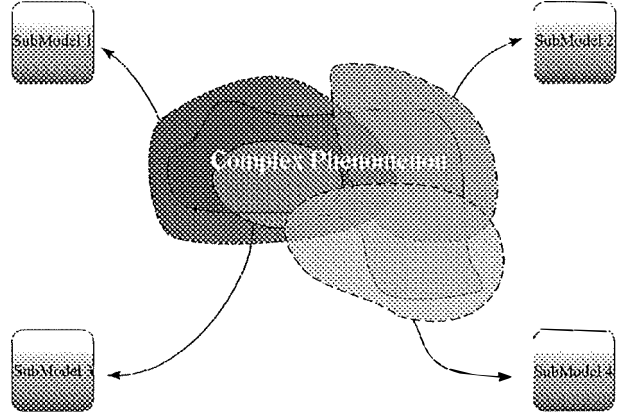


Fig. 1 Distribution of a process in a set of sub processes.

A model can be represented with a function f , in such a way a sub-model w of f too can be represented with a function, in this case with f_w .

Each f_w is the result of a distribution operation D_p of f according to the principium p .

$$D_p(f) = f_1, f_2, \dots, f_w$$

$$f_w : \mathcal{R}^{n_w} \rightarrow \mathcal{R}^{m_w}$$

$$\mathcal{R}^{n_w} \subseteq \mathcal{R}^n, \mathcal{R}^{m_w} \subseteq \mathcal{R}^m$$

Every model is an exemplification of the description we can obtain from the phenomenon. Thus every model describes a specific simplification of reality. The possibility of distributing a model guarantees some advantages: the first one is the utilization of the models with high resolution and a lower simplification degree, because these models are composed of many sub-models each one for a specific context. Another advantage is the to utilization of parallels models at the same time, in such a way it is permitted the coexistence of different models describing the same phenomenon or different phenomena.

The relations among the sub-models play an important role in the control process.

2.2 Control activities distribution

Another (orthogonal) role the distribution plays in control is connected to the functional distribution of the control activities that can be applied to the design of controllers. The controller can be itself a distributed entity. Here we adopt a distribution of the functions in a controller that is inspired to the classical robotic approach [6]. As shown in fig. 2, we consider a controller structured in three main steps: *knowledge extraction*, *decision making*, and *plan generation*.



Fig. 2 Distribution of control process in two main processes: decision making and plan generation.

The knowledge extraction step extracts high-level information from low-level data received from sensors. The parameters values it produces describe the state of controlled phenomenon. A decision is kind of desire, it expresses a desire and the importance of the desire for the system.

The decision making process does not consider the possible actions the system can perform. The decision making process is performed by a specific unit. Instead, the description of the actions the control system can perform is introduced in another specific unit. Thus the process control results performed in three phases: the first one is related to the perception of the current state of the phenomenon to be regulated (knowledge extraction), the second one related to the generation of the desirable new states (decision making), and the last one related to the actions that are actually performed to reach the desirable states.

Decisional making system uses the model of the process; plan generation system uses the model of actions that the control system can perform.

2.3 A distributed control system based on multi agent system

We have adopted for our system the both two kinds of distribution, in an architecture based on the modern paradigm of distributed artificial intelligence called *multi agent system* [10]. Some examples in literature of such systems can be: [9,8,3,7]. Especially, the context of our system is the physiologic processes control. Our system has been called: *Anthropic Agency* [4].

We distribute control process in a set of agents: *extractor agents* make knowledge extraction, *decisional agents* make decisions, *actuators agents* make action plan. We distribute a complex model in a set of sub-models, each one embedded in a specific decisional agents. Meanwhile extractor agents do not interact among them, the other two classes of agents do that; in a

such a way only decisional and actuator agents are interesting for us in order to describe the control process.

Thus we can identify two agents' networks, the first one of decisional agents and the second one of actuators agents; the decision making is the result we can obtain by the interactions among the agents of the first network, the action plan is the result we can obtain by the interactions among the agents of the second network.

Hence, the interactions establishing in the two networks determine the behavior of control system.

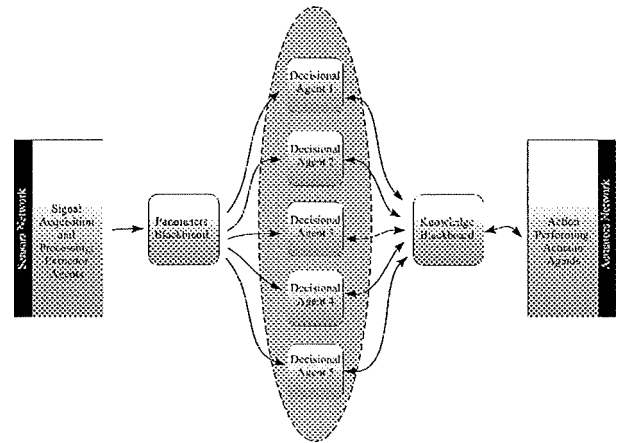


Fig. 3 Anthropic Agency architecture.

In this report we are interested only in the decision making process, because it involves the sub-models of the physiological process.

2.4 Decision making in Anthropic Agency

The architecture sets just the protocols of the decision making results according to an Agent-Oriented approach, the principia the agents can adopt to perform such process are absolutely free. The decision making process must generate a list of decisions for each variables of entire complex model. Thus the mechanism adopted by decisional agents to produce their decisions and the interaction paradigm adopted to perform the data fusion process of partial decisions can be modified according to the specific applications.

A decision for a variable is a pair: a desired state and the corresponding weight. The weight is a measure of how much the current state is away from optimum and, thus, of how much the decisional agent "wants" to reach the desired state.

In our approach each decisional agent makes an analysis of the current state of the sub-process it controls, and communicates its decisions about that.

Every decision is communicated to an element, named *equalizer*, that makes the data fusion using a specific algorithm [2]. The result of the equalizer process is a set of decisions. These decisions are communicated to the problem solver to obtain the action plan.

3 Physiological model

Physiological models represent a particular class of processes models. Physiological models are the mathematical transcription of the knowledge about a specific human body biological process. Physiological processes include, for instance, glucose regulation, blood pressure regulation, heart frequency regulation, etc...

Human body phenomena present three types of complexity. They are complex to isolate, to estimate and to translate in a model.

The first type of complexity is related to the fact that the effects of each physiological process activate many side processes; hence it is very difficult for us to isolate the process we want to study. Notwithstanding, the isolation of the area of interest is the first step to try to understand the behavior of the process. We can not offer any aid at this level.

The second type of complexity is related to the estimation of the process behavior. According to the classical identification techniques, some substances are introduced in the human body to generate some disturbs in the metabolic system in order to study the response behavior of the process. So it is possible to study the behavior of a specific part of the body compared with the variations of a physiological signal that represents an input of the studied system. In this context, input signals are hormones, proteins, molecules as glucose, and so on. These substances may be toxic for human metabolism; in such a way it is possible to use just a few kinds of substances in small doses. In addition to that the measure of any substance inner the body is very complex to be performed. Hence it results simpler for us to study just one aspect of the process by one, and not the entire process just in one time. In such a way it results more easy to obtain partial models of a process than an unique complex model of such process; this fact emphasizes the decentralization approach we have adopted.

The last type of complexity is related to the not-linearity and to the variability of the physiological processes behavior. This makes the modeling of system and, later, the utilization of the model extremely complex. This fact emphasizes the adoption of the solution space as model representation. In addition we can get round the not-linearity problem in a way more, we can design many partial models each one with a specific existence field, in such a way each model has just a specific range of the values of its variables in which it works [5]. Hence we can obtained a set of models each one for a

specific state of the process according to the decentralization approach.

3.1 Variables classification

In a model it is possible to identify variables and parameters. The main difference is that the variable value depends on the time while the parameter value is a constant over the time.

Variables can be classified according to the role they play in the model. Some variables constitute the external inputs of the model, such hexogen variables are not affected by the behavior of the simulated process; the others variables are inner to the model and their values describe the behavior of the simulated process. For instance, the insulin concentration and glucose concentration depends on the glucose regulation metabolic processes, in other words the value of these variables represents the effects of the process behavior; the meal adsorption rate is not affected by the behavior of the process, in fact it constitute an input of the model. We call variables directly determined by the simulated process behavior *internal variables*; instead we call the variables that are not affected by the simulated process behavior *external variables*.

The model can be represented as a function f with different spaces as domain and codomain.

$$f: \mathcal{R}^n \rightarrow \mathcal{R}^m$$

$$f: \underline{x} \mapsto \underline{y}$$

Each input vector \underline{x} is constituted by external variables $x_{e_i}(t)$, internal variables $x_{i_j}(t)$, and parameter p_z . According to this classification it is possible to group the variables that constitute the vector \underline{x} in three sub-vectors.

$$\underline{x} = ((x_{e_1}(t), \dots, x_{e_k}(t))_p, (x_{i_{k+1}}(t), \dots, x_{i_{k+k}}(t))_p, (p_{h+k+1}, \dots, p_n))$$

Each output vector \underline{y} is composed of the variables determined by the behavior simulated by the model, which are the internal variables $x_{i_j}(t)$.

$$\underline{y} = (x_{i_1}(t), \dots, x_{i_k}(t))$$

In a discrete time representation a state is the collection of values of variables at that time instant. The new state is calculated by using the values of the variables at the previous time step and the new disturbs at the current time interval.

In a discrete time representation the length of the time quantum is very important, especially for derivative or integrative variables (that, in a discrete time context,

become the difference or the sum of the variables values at two successive instants of time, respectively).

3.2 Physiological process variability model

In this section our focus is on the variability of a physiological process. We can identify two types of variability: the people variability and time variability. The first one kind of variability: the same physiological process may have a different behavior in different people. The second one kind of variability: the same physiological process may have, in the same person, a different behavior at different instants of time.

It is possible to describe the variability of a process with an unique model by assigning different values to some specific parameters.

Each process can be represented with a model, an input signal and an output signal. When the output signals obtained by using the same input signals changes during the time, it is possible to identify the effects of variability of the system. The presence of a pathology inner to a body causes a variation of the physiological process behavior compared with the healthy process one. Measuring by sensors the output of the process it is possible to identify some alteration of the current process behavior, if the entity of the variation of the process response is higher than an assumed threshold, we consider such process pathologic.

Higher the variation of some parameters higher the variation of the response of the system. We call *parameters configuration* the vector of values of the parameters of interest. Finally we associate a specific response of the system to each specific parameters configuration. We call pc_i the parameters configuration, according to the indices used previously:

$$pc_i \in \mathfrak{R}^{n-k-h}$$

$$pc_i = (p_1, p_2, \dots, p_{n-k-h})$$

In the case of physiological processes, the values of just a reduced set of parameters may change and that permits to simulate all the pathologies that can be found in nature. In such a way we describe the pathology as a variation of just some specific parameters that we call *pathologic parameters*. According to what we said before, we use the terminology *pathologic parameters configuration* pc_p to indicate the vector composed of the pathologic parameters. In addition we call *healthy parameters configuration* pc_h the configuration that corresponds to the model of a healthy man.

$$pc_h \in \mathfrak{R}^r \subseteq \mathfrak{R}^{n-h-k}$$

$$pc_p \in \mathfrak{R}^r \subseteq \mathfrak{R}^{n-h-k}$$

3.3 Particularity of physiological model

A dynamic process has a set of input signals and a set of output signals. Both input signals and output signals are curves in time.

The particularity of a biological process is that it is characterized by having similar input signals during the time; in fact the profiles and shapes of curves of the input signals of a biological process can be distinguished by very few characteristics. Although the same process may have a lot of different inputs in nature, it is possible to identify for every different input a reference shape. For instance, in the glucose-insulin metabolism model we have adopted, two inputs have been considered: meal adsorption and physical activity; for each of these inputs there is a reference input curve. In our experience, we have found that the mean characteristic of the differentiation is due to the amplitude of the curve. In figure 4 we report the curves of the meal adsorption rate.

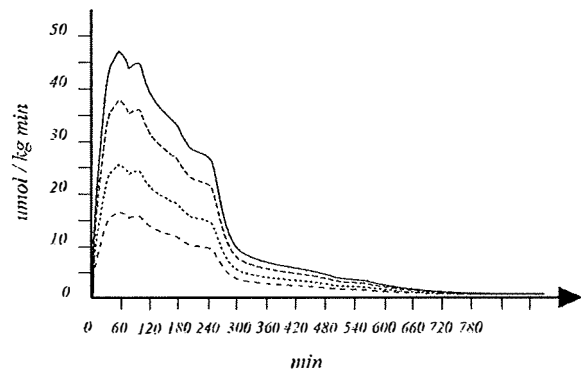


Figure 4 Meal adsorption rate curves.

The result is: the output signals of a biological model have usually similar properties in time exactly as input signals; with regard to output signal context, the mean difference we have found is due to the amplitude of the curve.

Thus we obtain a specific output curve for each input signal we introduce; according what we said before the curves represent the solutions of the model. If we consider the space of the solutions, we note that the output curves, obtained from the model with a particular parameters configuration and a particular input, reside just in a limited region and they do not occupy the entire space. If we change the values of the pathological parameters configuration we note that the regions of space of the solutions we have obtained change in comparison with the previous case. Every parameters

configuration generates a specific own region of space. Thus we can introduce the concept of *reachable space* for a specific parameters configuration. A space region is a reachable space for a parameters configuration if all points of this space region belong to the solutions curves of the model with that particular parameters configuration. Every parameters configuration has an own reachable space different from others.

We are interested only to the regions generated by healthy parameters configuration and by all pathological parameters configurations. We are not interested of regions generated by parameters configurations concerning non-pathological parameters.

3.4 Physiological processes analysis

Every input signal of the model generates only one output for each specific parameters configuration. So, each output signal of a model is bind to a specific parameters configuration and a specific input signal. We expect that parameters configuration of the model of current process is the measure of how this current process works in relation of the behavior taken as reference. The more the current parameters configuration is different from the reference parameters configuration, the more current system behavior is different from the reference behavior. The study of phenomena can not start from the study of parameters of its model representation because we can not measure them; in fact the problem is to estimate the current parameters configuration from current process behavior. Thus we have to try to estimate the parameters configuration from the process information we can measure by sensors.

4 Generation of the model representation

The generation process of the representation of the model presents three theoretical problems and one implementation problem. The theoretical problems are in order: to find a static representation expressive enough to describe a dynamic process, to describe the variability of such process, to give a semantic in order to relate the variability of such process with the degree of the pathology of such process. The implementation problem is due to the discretization of the variables; the discretization of the model is necessary to generate a finite and utilizable representation.

The kinds of information we use in the representation generation process are two: the set of equations of the model and a relation between the pathology degree and the variation of the process behavior.

4.1 Multidimensional space

We need a static structure in which the agent can find any possible interesting solution of the model we need, in addition, we need a set of standards to make the model representation to be Object-Oriented. We have chosen to adopt the solutions space. Such space is obtained by a discretization process. We remark that the space of solutions guarantee to us to not ignore the high-not-linearity of the physiological models.

Asides that space we need some elements that describes the meaning of such space, in other words these elements contain the semantic of the space. Moreover these elements with the multidimensional space constitute the Object. Such elements are: the names of the variables, the ranges of values for the variables, the discretization intervals, the scale factors of these intervals and the quantum of time discretization.

The range of values of a variable is the interval in which the variable values of such variable can be.

The discretization intervals of a variable are composed of the intervals in which the dimension associated to such variable has been divided into.

Each scale factor is constituted by a set of ranges of values for each discretization interval. The adoption of scale factors guarantee a better resolution grid in some regions of space than in others. The necessity of such structure can be motivated by the fact that some regions of the space are more interesting than others, so we want to describe better such regions than the others. Thus when the range of values of a variable is wide, it is often useless to have a fine grid for all values ranges, instead it is enough to have a fine grid just in some specific regions of space. Moreover the scale factors solve a part of the memory problems: a space with many and wide ranges occupies a large part of memory. Thus the utilization of these factors reduces the final model memory occupancy.

Before to explain the importance and the utilization of time quantum we need to introduce a classification of the variables that compose the state vector.

A generic state of the process can be represented by using a vector composed of variables values obtained from the processing of sensors signals. Such variables values can be divided in two groups: the first one composed of variables directly measured by the sensors, the second one composed of variables that are the result of the elaborations of the measured ones. The name of the first group is *measured variables*, the name of the second group is *calculated variables*.

For instance, glucose and insulin concentrations are measured variables; on the other hand the variation of glucose between two time instants is a calculated variable.

We discuss about the necessity of time quantum in the adopted model representation. The time quantum of

discretization process is information that is external to the model representation, but it is necessary for the correctness of agent activity. The quantum time information is important just for the calculated variables. For instance, if a model representation has been generated using a quantum of time equal to 30s, the agent need to receive the calculated variables generated by an algorithm according to 30s; if it receives data calculated at intervals of time different from 30s, the knowledge the agent infers utilizing such data could be wrong. In fact in discrete signal processing derivative and integrative processing become the difference and the sum respectively, in this way the sum or the difference of variable values in a temporal window of 60s will be different in comparison with a 30s window. It is obvious that if the adopted time quantum for the model representation and the time quantum of signal processing system are different the agent produces inexact results. Hence the representation of a model must have for each calculated variables its time quantum of discretization, in this way decisional agent informs signal processing system about that.

4.2 Solutions generation

We need at this point to introduce a description of the variability of the process. We consider a physiological model with a specific configuration of parameters as the reference model. The reference model represents the optimum behavior we desire the process accomplishes; in other words we desire that the process behavior will be properly describer by such model. A pathologic behavior can emerge in a real human being because of the corruption of some internal organs. We describe the presence of a pathological state by changing the values of some parameters.

We discuss the solution generation process.

A solution of the model is the response — of the model to input \underline{x} . A solution is, generally, a vector of curves, in our specific case, a discrete system, it is a matrix of the values of just each measurable internal variable. For each such variable the solution reports values during the time, according to the time quantum of discretization.

$$S = \begin{pmatrix} ix_1(t_1) & ix_1(t_2) & \dots & ix_1(t_T) \\ \dots & \dots & \dots & \dots \\ ix_s(t_1) & ix_s(t_2) & \dots & ix_s(t_T) \end{pmatrix}$$

The solution represents the dynamic response of the model at the specific input. The solution has a number of values for each variable equal to the length expressed in number of time elements of the response transitory.

We consider an external variable (see Section 3.1) and we make discrete the entire curve that constitutes such input variable. We generate one solution for every input

signal and for every parameters configuration pc of interest (see Section 3.3). We consider interesting a solution if it can be found in the reality. With a fixed parameters configuration pc , we introduced a set of input signals, as result of a discretization process of all possible input signals, and we record the obtained solutions. Then we changed the parameters configuration and restarted the process. Each solution lies in a specific region of the entire space, called reachable space (see Section 3.3); in this way the reachable space is a parameter that identifies the solutions in the multidimensional space.

The problem is to assign a specific representation to the multidimensional space of the specific reachable spaces obtained by the solutions.

4.3 Pathology degree

We need to introduce a relation between the pathology degree and the parameters configuration; such relation supplies the different solutions we have obtained with the semantic. The parameters configuration of the reference model is bind to the healthy person, so the more the current parameters configuration is different from the reference configuration deeper is the pathology degree. The representation of the pathology degree is given by a potential function. We assign a potential value to each solution obtained with a particular parameters configuration and a particular input. A lower value of the potential function means that the point corresponds to a healthy parameters configuration. Deeper the pathology degree higher the potential function value. In such a way each state of all the admissible solutions identify the pathology degree of the process that can generate such state.

In this way it is possible to identify whether the physical process we are monitoring works along healthy curves or pathologic curves and, in this last case, to estimate the pathology degree.

Formally, we call pc_h the parameters configuration of an healthy man, pc_p the current pathologic parameters configuration, and $P(S)$ the potential value related to solution S . In this way we obtain:

$$g(pc_h, pc_p) = P(S)$$

Potential values range is from 1 to 100, 1 means that current state is close to that of a healthy man model state, 100 means that current case is extremely pathologic.

The function g we have used in our experiments is following one:

$$pc_h = (p_{h_1}, \dots, p_{h_z})$$

$$pc_p = (p_{p_1}, \dots, p_{p_z})$$

$$f(pc_h, pc_p) = \alpha \cdot \prod_{i=2}^z \frac{p_{h_i}}{p_{p_i}}$$

The coefficient α is a normalization factor that forces the result value to be in 1 - 100 range.

Hence we can assign a potential value to each interesting solution of the model; this means that each reachable space in multidimensional space has a potential value.

In this way we have got a set of solutions and for each solution a potential value.

4.4 Data introduction to the multidimensional space

The last step is to introduce the reachable space of each solution in the entire multidimensional space.

The problem to be tackled is related to the previous discretization process. To make use of the possible all input curves and of all the parameter configuration values combinations is impossible, so we can not guarantee that every point of the multidimensional space is concerned by the solutions. In other words, some points of the multidimensional space may not be filled by generated solutions; consequently it is not possible to guarantee that every point of multidimensional space has a potential value.

At the beginning of the process we can consider all values of a solution and a pathology degree. For each time t_i , we have thus a state vector $(x_{i_1}(t_i), \dots, x_{i_j}(t_i))$ of the specific solution (see Section 4.2).

As already said, the values $(x_{i_1}(t_i), \dots, x_{i_j}(t_i))$ are related to the scale factors, and are transformed in to indices of the multidimensional space. Hence we have obtained a vector of indices that identify a cell of the multidimensional space.

$$SF(x_{i_1}(t_i), \dots, x_{i_j}(t_i)) = (index_1, index_2, \dots, index_j)_i$$

Initially we have an empty model that is an empty multidimensional space. An empty model is a multidimensional space where every point $(index_1, index_2, \dots, index_j)$ is initialized to the potential value corresponding to the maximum pathology degree. Thus the points of the space that are out every generated solution are considered equal to the extremely pathologic points.

The problem of filling the space could be easily solved by simply putting the potential value for every vector solution $(x_{i_1}(t_i), \dots, x_{i_j}(t_i))$. By doing that we would

face three major problems. The first one is that the model representation would be empty, except at some few states. The second one, which is related to the first one, is that a realistic model representation should take in account the continuity propriety around pathological states: a state with good pathological degree should not be close to a state with bad pathological degree. It would happen if we were putting directly the pathological degree at the coordinates obtained from the vector solution. The last one is related to the way an agent using the model representation should search for local minima. The agent would apply a gradient descent search algorithm, and it is clear that a model in which every point is at the maximum value except at few points would not be suitable for a such search.

To solve this problem we introduce the notion of *deflagration*. A deflagration associated with a state is a process in which we construct a space in the model around such state in which pathological degrees are getting worse as we leave the state. Deflagration process involves a specific region of space centered in the state, which dimensions are defined by a specific function.

Thus the introduction of a point in the multidimensional space can be represented in the metaphor of plain bombing: the empty model can be considered as a plain and each vector solution put in the model makes a deflagration in the plain.

The deflagration is characterized by three factors: the *range of deflagration*, the *shape of deflagration* and the *profile of potential* inner to the shape in function of the range.

We have an algorithm that generates an iper-rectangle (in a space with more than two dimensions) that defines the region of space in which the deflagration will be. The size of such iper-rectangle is described by the deflagration range. The deflagration range is a vector composed of the lengths of the sides of the iper-rectangle. The criteria we have chosen for the calculation of deflagration range is to consider that better is the pathological degree higher is the range, because it is realistic to think that around a state with a good pathological degree there should be a lot of such states. Thus a non pathological state has the maximum total range of deflagration and the worst pathological state has a total range of zero. The total range depends obviously on the size of the multidimensional space expressed in intervals number: if the range does not depend on the size of the multidimensional space, we could for example find a situation where each deflagration would cover the whole space.

The range \underline{R} of the deflagration is the result of the following function:

$$h(SV, PD, SS) = \underline{R}$$

Where SV is the state vector, PD is the pathology degree of the SV , and SS is the size of space.

The deflagration shape defines what regions of points inner to the iper-rectangle contain the deflagration. Hence it is possible to utilize the preferred shape for every specific model.

The profile of the potential is the curve of the hollow generated by the deflagration. It is a function that defines the potential value of the points inner to the deflagration shape. The potential curve inner to the deflagration shape will have a minimum value close to the center of deflagration and a maximum value on the border of the deflagration region. The potential curve is a function that depends on the deflagration range, the deflagration shape, the coordinates of such point in relation to the deflagration center, and on the potential value of the deflagration center.

$$k(index_1, index_2, \dots, index_n) = (R, SV, PD, S, (index_1, index_2, \dots, index_n))$$

By default, we have considered the iper-rectangle as an iper-cube, the deflagration shape as a sphere centered in the source of the deflagration, and the potential of a point of the deflagration as a function of the distance between such point and the center of the deflagration.

At this point we discuss about the conflicts that may be generated. The only conflict concerns the assignment of the potential values to a point. Different deflagrations may assign different values to the same point. It is necessary to define some criteria to solve such conflicts.

Generally we consider for every point the assignment with the minimum value of potential, but we consider some exceptions.

We call *real point* the points expressed by the solutions. A real point value can not be changed by a deflagration, but just by another real point with a lower potential value, according to previous principium.

5 Results

We have tested the algorithm using, in our application, the model of glucose-insulin metabolism. We have generated a representation of this model with two kind of input: meal adsorption curve and physical activity.

We have considered two multidimensional spaces, the former constituted by the following variables: glucose, insulin and variation of glucose between two instant of time, this multidimensional space has been used for the solutions obtained by using meal adsorption input; the latter constituted by the following variables: glucose, insulin and physical activity, this multidimensional space has been used for solutions obtained by using physical activity input.

Spaces descriptions are the following.

Space Glucose, Insulin, Δ Glucose (GI Δ G) has the following range:

- Glucose: 60 – 300 mg / dl
- Insulin: 0 – 500 pmol / l
- Glucose: -60 – 120 dg / dl (30 s)

In addition these variables have the following number of intervals:

- Glucose: 100
- Insulin: 100
- Glucose: 60

All intervals ranges are equal to a constant.

Space Glucose, Insulin, Physical Activity (GIA) has the following range:

- Glucose: 60 – 130 mg / dl
- Insulin: 0 – 60 pmol / l
- Physical Activity: 0 – 60 pmol / l

In addition these variables have the following number of intervals:

- Glucose: 60
- Insulin: 60
- Physical Activity: 30

All intervals ranges are equal to a constant.

For all tests we have used the native language compiled version of Java classes; in this way the requested time for calculation has been reduced about three times.

All tests were performed on a Pentium III 850 MHz.

5.1 Test A

We have considered only healthy solutions.

The solutions used for the multidimensional space GI G were in number 50, and each solution was composed by 1000 time instants. So the number of points introduced in the space was 50*1000, although many among these points are repeated.

The solutions used for the multidimensional space GIA were in number 50, and each solution was composed by 100 time instants. So the number of points introduced in the space was 50*100, although many among these points are repeated.

The generator had the following parameters.

The shape of the deflagration is a sphere, and the range is so calculated:

$$D = 0.2 \cdot I_{\min} \cdot \left(1 - \frac{P}{100}\right)$$

I_{\min} is the minimum number of discretization intervals of the variables used in the multidimensional space.

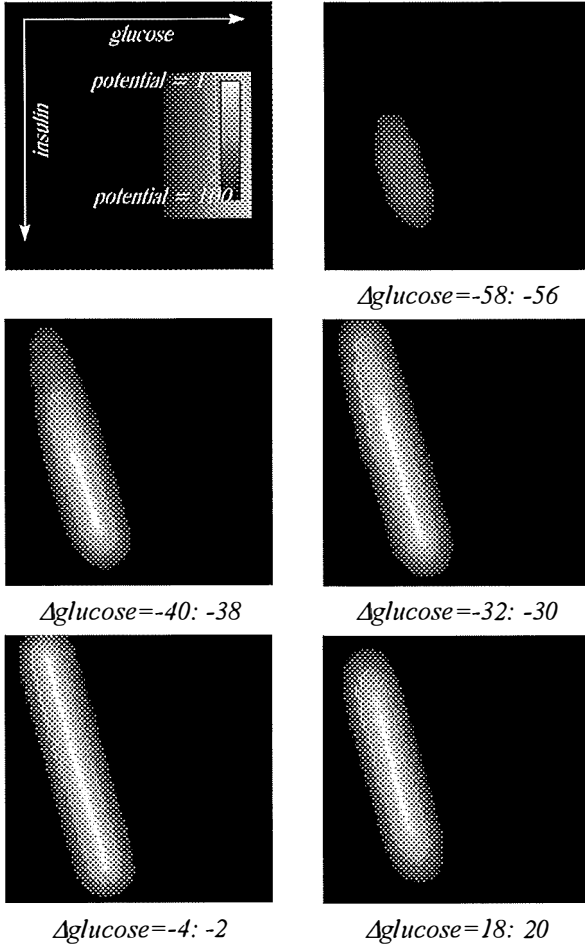


Fig. 5 Slices of Insulin-glucose planes at different $\Delta\text{glucose}$ values.

The potential curve along the range of sphere is so calculated:

$$P(\underline{x}) = \frac{D(\underline{x})}{D_{\max}} \cdot (100 - P_c)$$

(\underline{x}) is a point inner to the deflagration, $P(\underline{x})$ is the potential value of point (\underline{x}) , $D(\underline{x})$ is the distance between the point (\underline{x}) and the deflagration center, P_c is potential of deflagration center, D_{\max} is the maximum deflagration range.

Some slices of the GI G model representation, obtained using our generator system, are shown in figure 5. In this case deflagration is a sphere with a maximum range equal to 11.

Requested time for calculation is about 5 min.

Some slices of the GIA model representation, obtained using our generator system, are shown in figure 6. In this case deflagration is a sphere with a maximum range equal to 5.

Requested time for calculation is about 8 s.

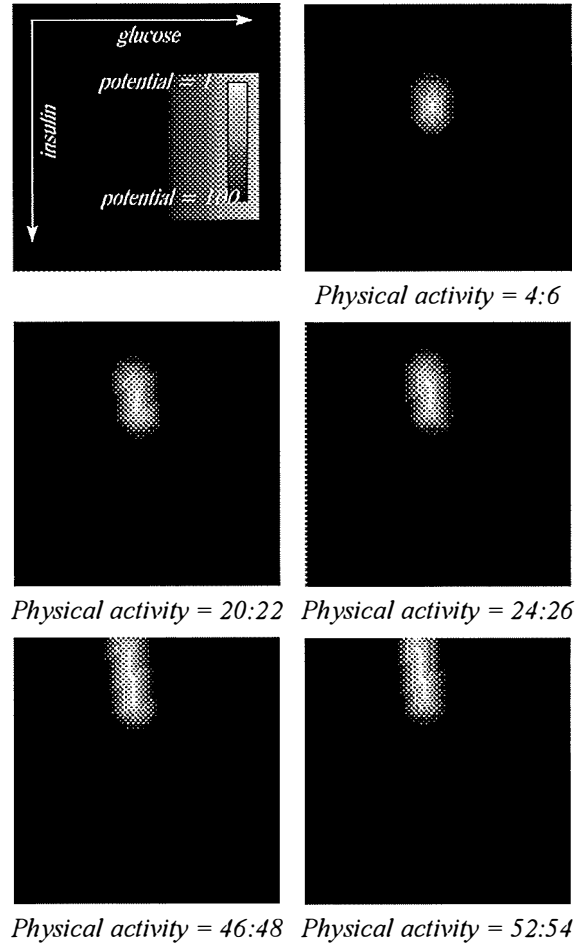


Fig. 6 Slices of Insulin-glucose planes at different physical activity values.

5.2 Test B

We have considered only healthy solutions.

The solutions number of both GIA case and GI G case is equal to Test A.

The generator had the following parameter.

The shape of the deflagration is a sphere, and the range is so calculated:

$$D = 0.2 \cdot I_{\text{mean}} \cdot \left(1 - \frac{P}{100}\right)$$

I_{mean} is obtained by the mean between the minimum number of discretization intervals of the variables used in the multidimensional space and the max number of discretization intervals.

The potential function inner the hollow deflagration is the same that is used in Test A.

Some slices of the GI G model representation, obtained using our generator system, are shown in figure 7. In this case deflagration is a sphere with a maximum range equal to 15.

Requested time for calculation is about 11 min.

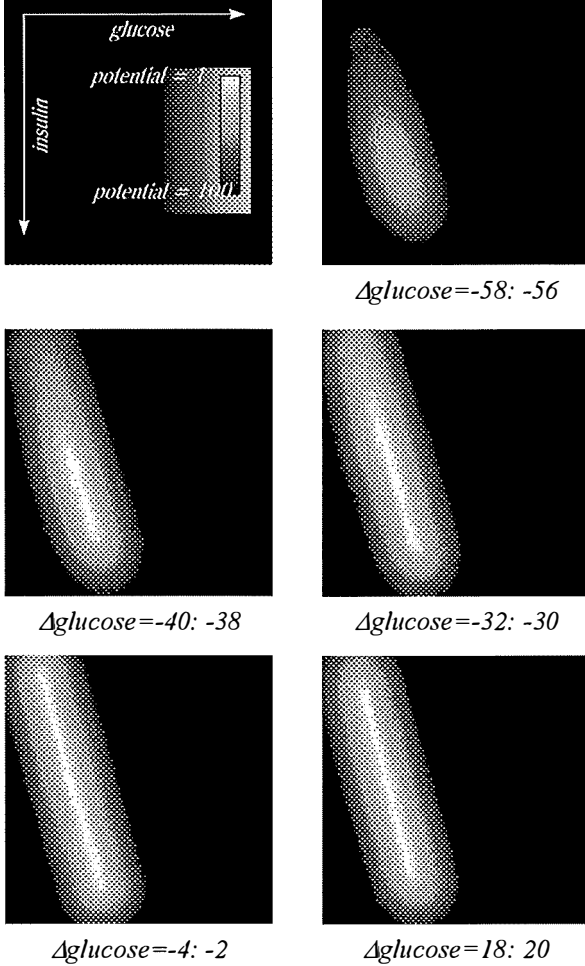


Fig. 7 Slices of Insulin-glucose planes at different $\Delta\text{glucose}$ values.

Some slices of the GIA model representation, obtained using our generator system, are shown in figure 8. In this case deflagration is a sphere with a maximum range equal to 8.

Requested time for calculation is about 15 s.

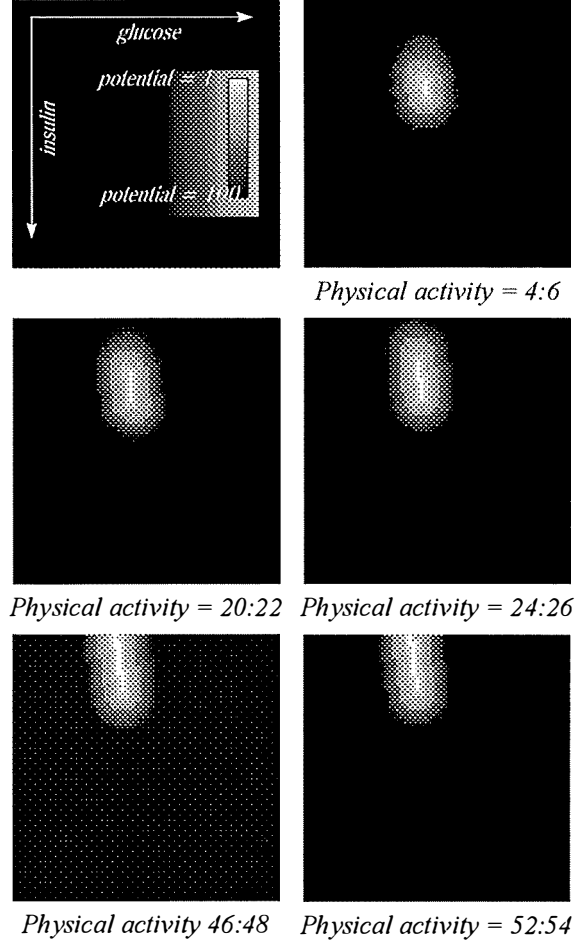


Fig. 8 Slices of Insulin-glucose planes at different physical activity values.

5.3 Test C

We have considered only healthy solutions

The solutions number of both GIA case and GI G case is equal to Test A.

The generator had the following parameter.

The shape of the deflagration is a sphere, and the diameter is so calculated:

$$D = 0.2 \cdot I_{\max} \cdot \left(1 - \frac{P}{100}\right)$$

I_{\max} is the minimum number of discretization intervals of the variables used in the multidimensional space.

The potential function inner the hollow deflagration is the same that is used in test A.

Some slices of the GI G model representation, obtained using our generator system, are shown in figure 9. In this case deflagration is a sphere with a maximum diameter equal to 19.

Requested time for calculation is about 20 min.

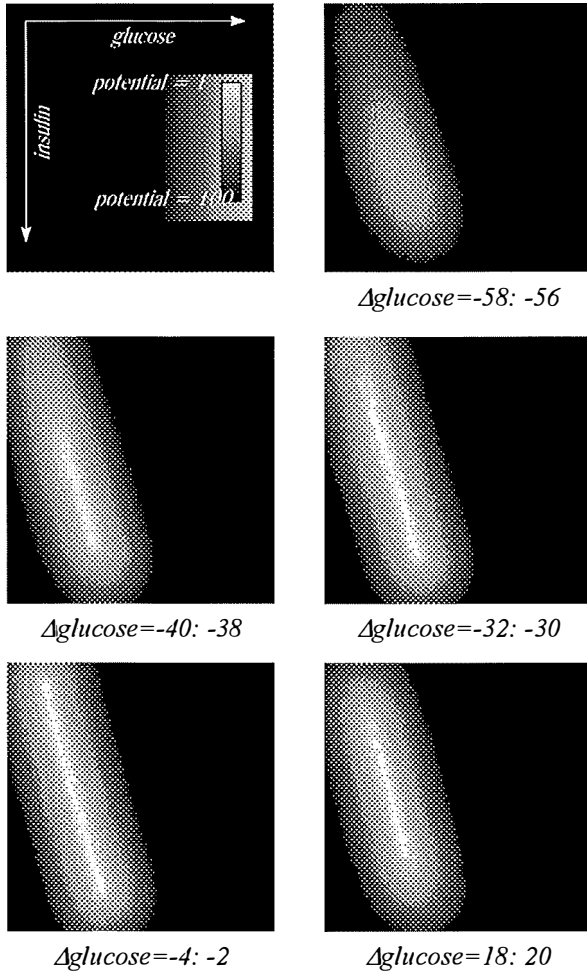


Fig. 9 Slices of Insulin-glucose planes at different $\Delta\text{glucose}$ values.

Some slices of the GIA model representation, obtained using our generator system, are shown in figure 10. In this case deflagration is a sphere with a maximum diameter equal to 13.

Requested time for calculation is about 45 s.

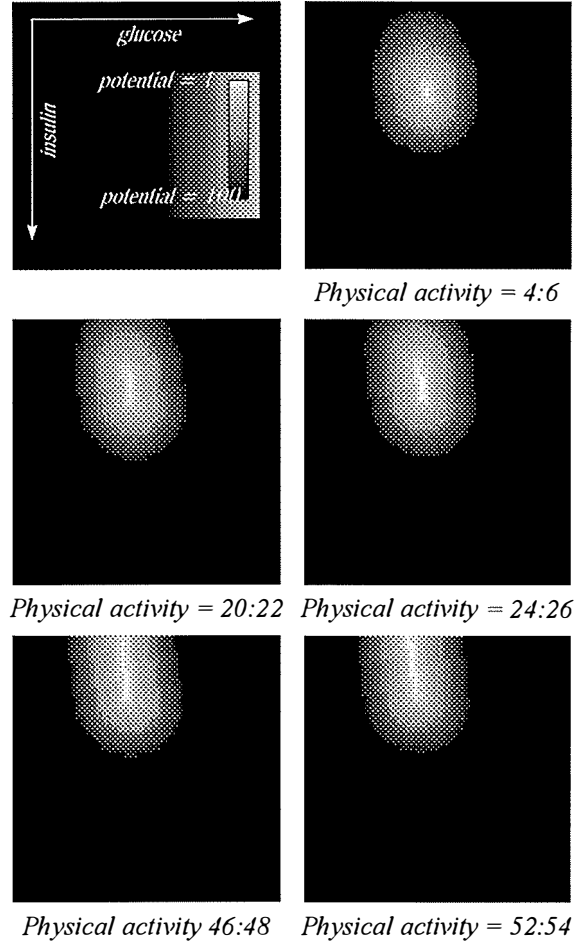


Fig. 10 Slices of Insulin-glucose planes at different physical activity values.

5.4 Test D

This test was performed with the same parameter of Test A, the unique difference is that, here, we have used only one status in four instants of time. So we have used a reduced number of data to generate the multidimensional space.

Some slices of the GI G model representation, obtained using our generator system, are shown if figure 11.

Requested time for calculation is about 4 min.

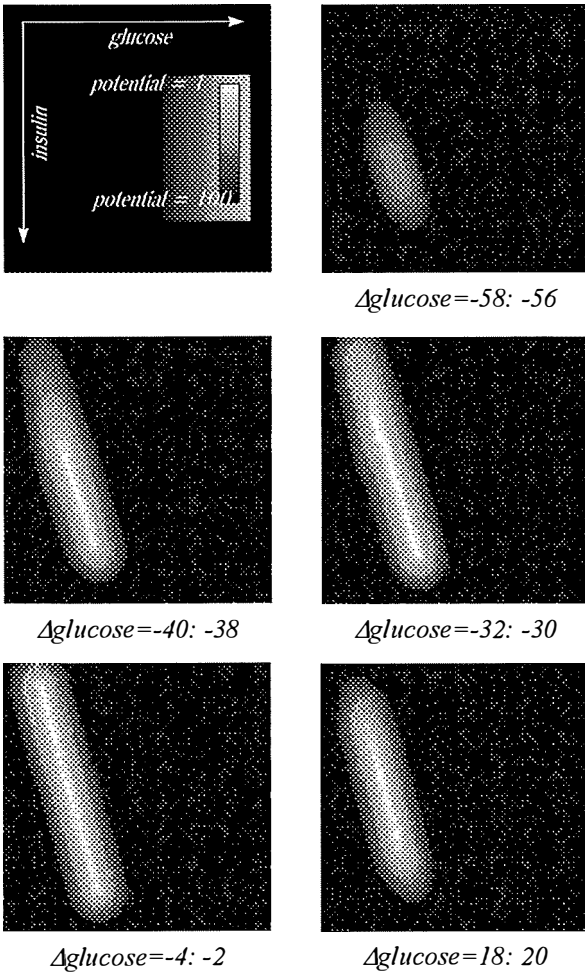


Fig. 11 Slices of Insulin-glucose planes at different $\Delta\text{glucose}$ values.

Some slices of the GIA model representation, obtained using our generator system, are shown in figure 12.

Requested time for calculation is about 7 s.

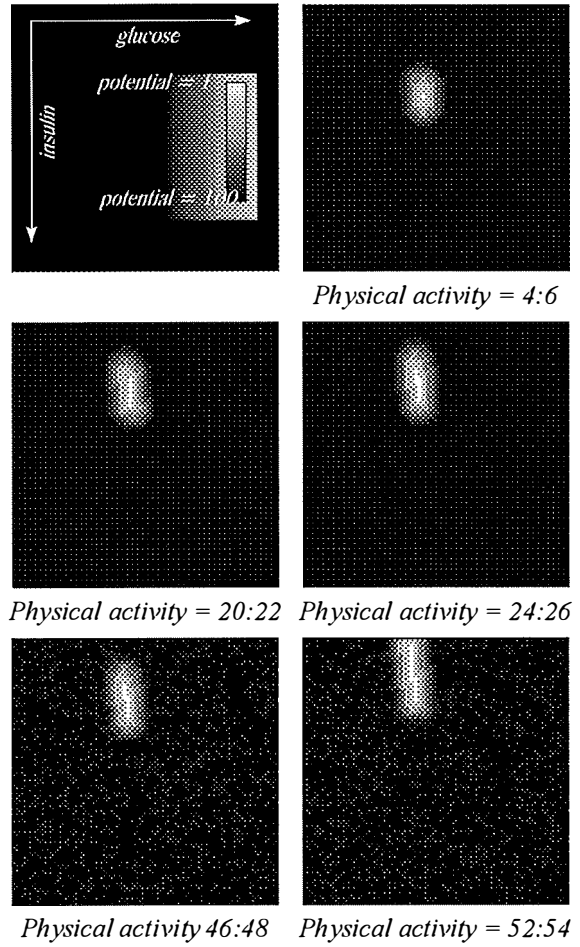


Fig. 12 Slices of Insulin-glucose planes at different physical activity values.

6 Conclusion

In this report we have proposed a theory for a new kind of models representation, that is obtained by the fusion of the information about the physical model equations and the information about the effects of the model parameters variation; we have proposed a data structure Object-Oriented for the proposed representation; finally we have proposed a package of algorithms to perform the generation of such representation.

6.1 Deflagration

We have adopted the deflagration approach as countermeasure to discretization effects: the impossibility to guarantee that each point of the multidimensional space has a potential value. The utilize of a fine grain of discretization permits to have more accurate values, but, in the other hand, it makes that the generation process of the multidimensional space takes hours and hours and it slows in particular

the multidimensional space filling process. In any case, every adopted discretization scheme for the parameters and the input curves can not guarantee a priori that some points of the multidimensional space will be missing or potential empty point. Moreover the continuity and uniformity of the multidimensional space are fundamental properties for the performing of decisional agent reasoning about that. Thus the deflagration approach permits to have homogeneous regions of potential, without any sort of discontinuity. The larger deflagration range the smoother and the more homogeneous the result space, but the lesser the precision of the representation. The smaller deflagration range the lesser homogeneous the result space.

The objective of a deflagration process is to find the optimum compromise of the following elements: uniformity of space, precision of representation, requested calculus time.

6.2 Time of calculus

The time requested for a generation of the space is depending on the size of dimensions of the space, on the numbers and the length of the solutions and on the parameters inner the space generation algorithm, for instance: the range of the deflagration hollow, the shape of the hollow.

The utilize of space with a fine grid and a large number of dimension slows strongly the generation process. We report the following proof: in our tests space GIA has always been generated faster than space GI G. The memory occupancy, we have measured, for a space with the dimensions of the order of the space we have adopt in our tests, is about one or two Mbytes.

The algorithm of the entire deflagration process that requires the most relevant part of calculus time is the algorithm that generates the deflagration and inserts the hollows in the multidimensional space. The larger the range of deflagration the more the time requested. This fact is proved by Test B and Test C, where calculus time increases strongly respect Test A.

6.3 Redundant information

The solutions generated by the simulation contain a large quantity of redundant information, we have demonstrated such fact when we have obtained in Test D the same result of Test A, although we have used only one point out four points of the solution. Notwithstanding the utilization of just a part of points instead of all, does not reduces the calculus time. The utilize of only some points does not reduce the number of deflagrations, then the information contained by the solutions are redundant for the space filling. This fact is an adding proof of the fact that the algorithm that requires the most part of calculus time is the generation of deflagration algorithm.

6.4 Upgrades in future

The system we have proposed is not complete yet. Our thought is that in future it can be improved.

Model equation

The main element that we have not solved yet is the transcription by a specific software of model equations in a Java class or in something like that. We have defined just a set of Java interface to obtain a general system that can perform the simulation of a process. Meanwhile we have made a graphical interface to introduce elements that describe the space and that are useful to the construction of that; we have not a graphical interface yet to introduce the equations. Our desire is to build a translator that receives a set of equation defined in XML and generates an object defined in the adopted language (currently Java).

Calculus speed

For the realization of our system we have used Java language. This choice was caused by the fact that the simulation system we had and all Anthropos Agency was implemented using Java. Java is not a program language designed to perform calculus process; in fact the results we have obtained emphasize the necessity of a long time to perform the calculus. We have solved partially such problem by compiling the code in native language, and then using the executable files. In any case the utilization of MathLab or C++ would reduce significantly the calculus of time.

7 Appendix A: classes structure

We identify three main processes that compose entire generation process. The first is the generation of an empty multidimensional space according to a space description, the second is the generation of solutions curves by the model for every parameters configuration, and the last is the generation of potential function in the space using model solutions. A space filled by the potential function is said *colored* because it has a value in every point instead of null value.

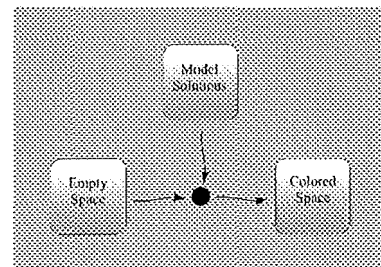


Fig. 12 Schematic model representation generation.

7.1 Space description

We identify two steps. The former step is the building of a description of the multidimensional space, the latter the utilizing of the description to build the space. *Space Description* is based on two descriptions: the description of the variables, the description of scale factors for each variable.

The variables to describe are the dimensions of multidimensional space; each variable corresponds to a dimension of the space. It is necessary to introduce for each variable: the name of variable, the range of values assumed by the variable – the minimum and the maximum values –, and the number of discretization intervals. This information is stored in *VariableDescription*.

For each variable we have a range of values of interest and the number of intervals in which the range has to be divided to. By default all intervals of the range are equal. It is possible, in any case, to set the length of each interval of the each range; this description is in *Scale Factors*.

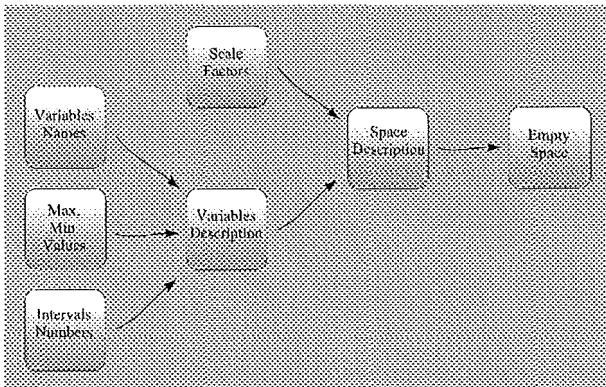


Fig. 13 Schema of space generation.

This information is stored in *ScalaDescription* class.

Both *VariablesDescription* and *ScalaDescription* are in *MBuilding* package. The package contains a graphical interface to introduce the description of variables and scales.

7.2 Solution generation

The objective of this system is obtaining a solution curve with the corresponding potential value.

The calculus of a solution is based on: a specific model, a specific input curve and a specific parameters configuration. The system considers a discrete number of input curves and a discrete number of parameters configurations. Each generated solution has a specific input curve and a specific parameters configuration.

The process of solution generator is performed by classes of *SimulationUnit.Simulation* package.

The mean class is *SolutionsGenerator* class. Any time such class is invoked, it returns a new solution.

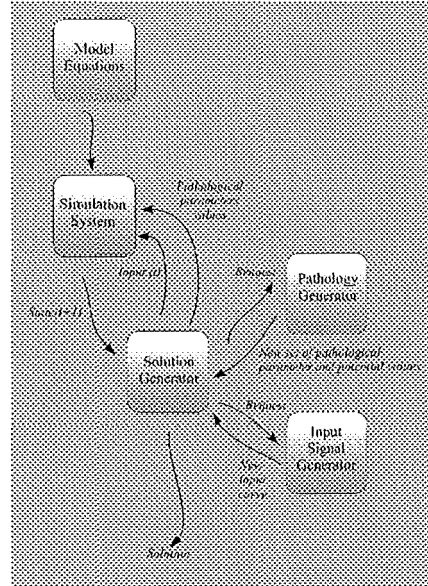


Fig. 14 Schema of space generation.

Simulation

Contains code *Java* transcription of equations of the model and it is the real simulation system. The model representation is made by the use of variables, parameters, and relations among these. Variables are array of double, and the index is time, parameters are double values, equations are methods.

Inputs of this system are external variables values and pathologic parameter of current solution. Output is a vector with variables values of next time interval.

We have used a sub class that includes all the variables of the model, *SystemParameter*.

InputSignal

Contains *Java* code transcription of curves that describe the input of *Simulation* class. These curves are parameterized respect to one or more indexes. One specific curve corresponds one set of indexes. *Input Signal Generator* receives the set of index as input, and generates the curves as output. *InputSignal* is, in reality, a *Java* interface, this fact permits to define a generated structure for all possible input curves.

PathologyGenerator

Contains the description of pathology: the pathological parameters and the relation between their variations and

the corresponding pathology degree. A pathology degree corresponds to a particular parameters configuration and then to a potential value. For each pathological parameter is fixed a priori a range of values and a discretisation. This class receives the request for a new pathological parameters configuration, and it transmits such configuration and the corresponding potential value. Currently PathologyGenerator is integrated with SolutionsGenerator class.

SolutionsGenerator

Each dynamic model with a specific input signal has a specific output; class SolutionsGenerator generates such output. A *Solution* is a structure that contains the values of variables of interest in relation of time and the value of potential of current output.

It asks a new input curves to InputSignal implemented class, it asks a new parameters configuration to PathologyGenerator class, and it communicates these to Simulation class and gets, from this last class, the new state. Holding a parameters configuration SolutionsGenerator invokes Simulation for all the length of input curve generated by InputSignal. When data received from Simulation class end, SolutionsGenerator introduces in Solutions structure the potential value returned by PathologyGenerator.

7.3 Model Generator

In order to introduce reachable space in the model, we need to accomplish two steps, as identified in Section 4.4: the first one takes the vector solution and the corresponding pathology degree from SolutionsGenerator, the second one makes a deflagration around the point in the model. We have thus two distinct entities: *GenerateDynamicModel* and *Deflagration*.

We precise now the representational invariants in the model. An *empty point* is a point that has maximum pathological value but that has not yet been changed. An empty point in the model is represented by the value '-1'. A *real point* is a point that is that was given by SolutionsGenerator, thus it is a point that has not been created by a deflagration. Finally a point is a hollow point only if it has not the value '-1' and if it is not a true. A hollow point has thus been created by a previous deflagration.

As seen in Section 4.4, we need a way to parameterize the algorithm. This is achieved by building interface that specifies the function to implement. From Section 4.4, we know that we have four parameters to leave to the user. Here are these parameters :

FunctionRange

This interface defines the function that must calculate the total range for a new real point. The function to implement is called *calculateRange()*.

FunctionHollow

This interface defines the function that must give the potential for a new point in the deflagration. The functions to implement are *function()* and *isInTheRange()*.

RuleConflictDeflaHollow

This interface defines the function that should be applied whenever a conflict between a deflagration and a hollow occurs. The function to implement is *applyRule()*.

RuleTruePoint

This interface defines Boolean functions that modifies the behavior of the algorithm whenever a conflict between two real points occurs. The function to implement are *replace()* and *defla()*.

It is important to note that the algorithm owns its own default parameters : *FunctionRangeDefault*, *FunctionHollowDefault*, *RuleConflictDefault*, *RuleTruePointDefault*.

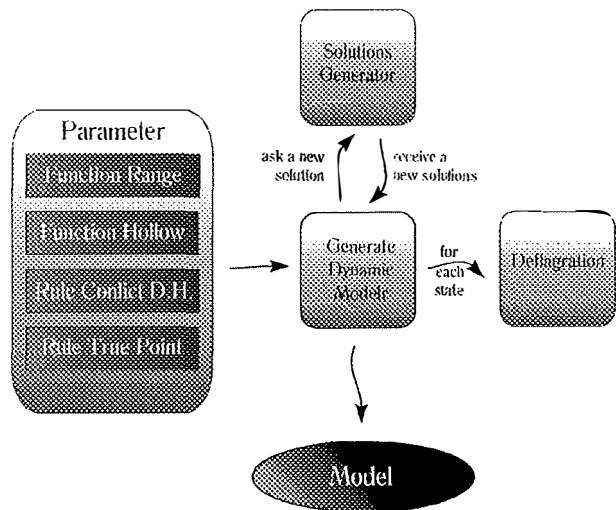


Fig. 15 Schema of space generation.

The whole process is carried on by *GenerateDynamicModel* and *Deflagration* and *GenerateDynamicModel* is parametrized by a series of interface as we have seen.

GenerateDynamicModel

For each solution and for each time, *GenerateDynamicModel* introduces real points in the

model. Before doing the core steps, it first checks that there is no real points conflict, and applies rules from *RuleTruePointDefault* if it is the case. If it is not a real point, it adds to a list that keeps track of real points in the model. Then it does the core steps by calculating the range with the parametrized function of *FunctionRange*, initializing the class *Deflagration* and launching the deflagration. These steps are reiterated until all curves have been done.

Deflagration

This class is used by *GenerateDynamicModel* in order to create the deflagration associated with a real point. The total range of the deflagration is passed by *GenerateDynamicModel* to *Deflagration*. *Deflagration* goes through every point in the range, avoiding real points, the real point associated with the current deflagration, and points that are outside the model. For every such point, *Deflagration* calls a parameter function in *FunctionHollow* which gives the function to be used in order to trace the deflagration. It then looks for conflicts with other points, i.e. hollow points. It applies the rules found in *RuleConflictDeflaHollow*, resolving the conflict. If it encounters a real point, it leaves the real point. When conflicts are resolved, it puts the final value in the model at current position. The deflagration for the current real point is done when every point in the range has been done.

8 Appendix B: classes code

We report here part of implemented code for any package.

8.1 MBuilder package

VariablesDescription class contains the description of a single variables (sub class *VariableDescription*), and a Vector of variables, *VariablesDescription* itself.

```
public class VariablesDescription
{
    public VariablesDescription (int dimensions)
    {
        this.dimensions = dimensions;
        this.Variables.setSize(dimensions);
        for (int i = 0; i < dimensions; i++)
        {
            this.Variables.setElementAt
                VariableDescription (i);
        }
    }

    public class VariableDescription
    {
        public VariableDescription ()
        {
            this.NameVariable = new String ();
            this.CorrespondenceVariable = new String ();
```

```
            this.MaxVariable = 0;
            this.MinVariable = 0;
            this.Intervals = 0;
        }

        private String NameVariable;
        private String CorrespondenceVariable;
        private double MaxVariable;
        private double MinVariable;
        private double Intervals;
    }

    private int dimensions;
    private Vector Variables = new Vector ();
}

ScalaDescription is initialized with an instance of
VariablesDescription, in this way ScalaDescription
builds a default Scala. In the default generated Scale all
range of a specific variable are equal.

public class ScalaDescription
{
    public ScalaDescription (VariablesDescription temp)
    {
        this.vardescr = temp;
        this.dimensions = this.vardescr.getDim ();
        this.Scala.setSize (this.dimensions);

        for (int i = 0; i < this.dimensions; i++)
        {
            double minimum = this.vardescr.getVariableMin (i);
            double maximum = this.vardescr.getVariableMax (i);
            int positions = (int) this.vardescr.getVariableInt (i);
            double coeff = (maximum - minimum) / positions;
            Vector vtemp = new Vector();
            vtemp.setSize (positions + 1);
            for (int y = 0; y < positions + 1; y++)
            {
                vtemp.setElementAt (new Double (minimum +
                    y * coeff), y);
            }
            this.Scala.setElementAt (vtemp, i);
        }
    }

    private int dimensions;
    private Vector Scala = new Vector ();
    private VariablesDescription vardescr;
}
```

8.2 SolutionGenerator package

A Solution is a vector of a set of variables during the time and a potential value. Solutions class is a set of Solution.

```
public class Solutions
{
    public Solutions(int solutionsNumber, int variablesNumber,
        int intervalsNumber)
```

```

{
    this.intervalsNumber = intervalsNumber;
    this.variablesNumber = variablesNumber;
    this.solutionsNumber = solutionsNumber;
    this.SSstack.setSize(solutionsNumber);
    for (int i = 0; i < solutionsNumber; i++)
    {
        this.SSstack.setElementAt(new
Solution
(variablesNumber, intervalsNumber), i);
    }
}

public class Solution
{
    public Solution (int variable, int lenght)
    {
        this.Sstack.setSize(variable);
        for (int i = 0; i < variable; i++)
        {
            this.Sstack.setElementAt(new
Variable(lenght), i);
        }
    }

    public class Variable
    {
        public Variable(int dim)
        {
            this.Vstack.setSize(dim);
            for (int i = 0; i < dim; i++)
            {
                this.Vstack.setElementAt(new
Double(0), i);
            }
        }

        private Vector Vstack = new Vector();
    }

    private Vector Sstack = new Vector();
    private int potential;
}

private int solutionsNumber;
private int variablesNumber;
private int intervalsNumber;
private Vector SSstack = new Vector();
private ScalaDescription Scala;
}

```

In SystemParameter is reported all variables used in the Simulation.

```

public class SystemParameter
{
    public SystemParameter()
    {
    }

    public double insulin;
    public double glucose;
    public double utilization;
    public double secretion;
}

```

```

    public double production;
    public double degradation;
    public double excretion;
    public double dglucose;
    public double physicalactivity;
    public double exogeninsulin;
}

```

Simulation defines all parameters of the simulation system, all variables as array of values in time.

Simulation implements *ProcessSimulation* interface.

We report the interface *ProcessSimulation*.

```

package SimulationUnit.Simulation;

```

```

public interface ProcessSimulation {

```

```

    public void Init();
    // this method initializes process variables to init value of
    stationary state

```

```

    public void reset();
    // this method resets all system variables, the value will be
    the init value

```

```

    public SystemParameter nextState();
    // this method returns the values of new state, it receives
    external variables values

```

```

    static int length = 0;
    int time = 0;
    double DeltaTime = 1;
}

```

We report now Simulation. The mean method of Simulation are: *Init* for initializing the system, *nextState* to obtain the state of process in time t+1, and a set of methods to set the parameter related to the pathology

```

public class Simulation implements ProcessSimulation
{

```

```

    public Simulation(double Dtime)
    {
        Init();
        this.DeltaTime = Dtime;
    }

```

```

    private void Init()
    {

```

```

        time = 1;
        p13 = 25.031 * this.DeltaTime;
        p15 = 186.475;
        BasalSecretion = this.BasalInsuline * this.p16;
        for (int i = 0; i < length; i++)
        {

```

```

            PhisicActivity = 0;
            ExogenInsuline = 0;
            Meal = 0;
            Insuline[i] = this.BasalInsuline;
            Glucose[i] = this.BasalGlucose;
            Secretion[i] = this.BasalSecretion;
            .....
        }
    }
}

```

```

}

public void reset()
{
    Init();
}

public SystemParameter nextState (double meal, double
activity, double exogeninsuline)
{
    this.Meal = meal;
    this.PhysicActivity = activity;
    this.ExogenInsuline = exogeninsuline;

    this.degradation();
    this.utilization();
    this.secretion();
    this.epaticproduction();
    this.excretion();

    this.Insuline[time+1] = this.Insuline[time] +
(this.Secretion[time] - this.Degradation[time] +
this.ExogenInsuline)*this.DeltaTime;
    this.Glucose[time+1] = this.Glucose[time] +
(this.EpaticProduction[time] - this.Utilization[time] -
this.Excretion[time] + this.Meal) * this.DeltaTime /
this.volumeGlucose;

    this.LastInsuline = this.Insuline [time+1];
    this.LastGlucose = this.Glucose [time+1];
    this.LastUtilization = this.Utilization [time+1];

    parameter.degradation = this.Degradation [time];
    parameter.glucose = this.Glucose [time];
    parameter.insuline = this.Insuline [time];
    .....

    if (time==0)
    {
        parameter.glucose = 0;
    }
    else
    {
        parameter.dglucose = this.Glucose[time] -
this.Glucose[time-1];
    }

    time++;
    return parameter;
}
.....

public void setDiabetic ()
{
    p13 = 0.1 * this.DeltaTime;
    p15 = 1;
    BasalSecretion = this.BasalInsuline * this.p16 * 0.7;
}
.....

private final static int length = 1200;
private int time = 1;
private double DeltaTime = 0.5;
private double LastInsuline = 0;

```

```

private double LastGlucose = 0;
private double LastUtilization = 0;

private double Meal;
private double ExogenInsuline;
private double PhysicActivity;

private double Insuline[] = new double[length];
private double Glucose[] = new double[length];
private double Secretion[] = new double[length];
private double EpaticProduction[] = new double[length];
.....
private double BasalInsuline = 26.31;
private double BasalGlucose = 5.1;
private double volumeGlucose = 154.52;
private double weight = 70;

private double p2 = 0.89 * this.DeltaTime;
//0.9498
private double p3 = 0.0000000000184 * this.DeltaTime;
private double p4 = 0.0032 * this.DeltaTime;
//0.0089
.....
private double HGRb = this.p8 * this.BasalGlucose;
private double p1 = (this.HGRb + (this.p2 *
this.BasalGlucose) + (this.p3 * this.BasalInsuline) + (this.p4
* this.BasalInsuline));

private double BasalUtilization = this.p8 *
this.BasalGlucose;
private double BasalSecretion = this.BasalInsuline *
this.p16;
private double BasalDegradation = this.BasalInsuline *
this.p16;
private double BasalGlucoseProduction = this.HGRb;

private SystemParameter parameter = new
SystemParameter();
}

```

InputSignal interface defines the standard methods for generating the entire input curve.

```
public interface InputSignal {
```

```

    public double[] inputCurve (double coeff);
}

```

Here we report SolutionGenerator, each methods is followed by its comments.

```
public interface SolutionsGenerator {
```

```

    public Solutions nextSolution();
    // this method returns the next solution to introduce in the
generator of representation of the model.

```

```

    public void Init();
    // this method initializes the description of the space of the
model and the description of the solution.

```

In particular: this method calls a graphical interface to set name of variables, the ranges, the intervals number, and the scale factor.

```
public Solutions extractSolution();
// this method contacts simulation class to generate a new
solution in relation of a new input curve and of a new
pathological parameter configuration.
In particular: the methods generates one solution for each
input curve in a healthy state, and one solution for each input
curve in any pathologic parameters configuration.
```

```
public boolean next();
// this method returns true if another solution exists, false if
the solutions are concluded.
```

```
public int getVariableNumber();
// this method returns the number of the variables that are in
the solution.
In particular: the number of dimension of the space to
generate.
```

```
public ScaleDescription getScale();
// this method returns the description of scale factor of the
variables of solutions.
```

```
public VariablesDescription getVariableDescription();
// this method returns the description of the space to generate.
In particular: name of the variables, ranges of values, interval
number.
```

```
public int getIntervalPerDimension(int dim);
// this methods returns the number of intervals of discretitation
for each dimension.
}
```

8.3 GenerateDynamicModel package

We recall that the aim of GenerateDynamicModel is to introduce reachable space in the model.

```
public class GenerateDynamicModel {

private ModelPrintGenerate model;
private Solutions solutions;
private SolutionsGenerator solutionsGen;
private int minimum;
private int maximum;
private int range;
private Vector truePoints;
private Vector order;
private int numberOfSolutions;
private FunctionRange funtionRange = new
FunctionRangeDefault();
private FunctionHollow functionHollow = new
FunctionHollowDefault();
private RuleConflictDeflaHollow rule = new
RuleConflictDeflaHollowDefault();
private RuleTruePoint ruleTruePoint = new
RuleTruePointDefault();

public GenerateDynamicModel()
{
```

```
System.out.println("Constuctor GenerateDynamicModel()
launched.");
this.truePoints = new Vector(0, 0);
this.model = new ModelPrintGenerate();
```

```
this.solutionsGen = new SolutionsGeneratorGLA();
this.solutionsGen.Init();
```

```
//Give the sizeOfModel to the default range function
this.funtionRange.putSizeModel(constructSizeOfModel());
```

```
//Construct an empty Model, filled with -1
System.out.println("Please construct a model filled with -1.");
```

```
this.model.constructModelFilled(this.solutionsGen);
}
```

```
/******
*** SET *****/
```

```
public void setMinimum(int min)
```

```
public void setMaximum(int max)
```

```
public void setFunctionRange(FunctionRange fctRange)
```

```
public void setFunctionHollow(FunctionHollow fctHollow)
```

```
public void setRuleConflictDeflaHollow(RuleConflictDeflaHollow r)
```

```
public void setRuleTruePoint(RuleTruePoint ruleTP)
```

```
/******
*** ALGORITHM ***
*****/
```

```
public void generateModel(String nameFile)
//This is the main algorithm.
//PRE : the first part of the system has been correctly set.
//POST : the resulting model has been written in nameFile.
{
```

```
//Set up the deflagration
Deflagration deflagration = new Deflagration();
this.setUpDeflagration(deflagration);
this.numberOfSolutions = 1; //Never used ?
int count = 0;
//ALGORITHM STARTS
while (solutionsGen.next())
{
```

```
    //Let's take the next solution.
    solutions = solutionsGen.nextSolution();
```

```
//calculate the range for this potential
```

```
int
r=this.funtionRange.calculateRange(this.solutions.getPotentia
l(0));
```

```
    //For each point at a given time, do
    for (int t=0; t<this.solutions.IntervalsNumber();
t=t++)
    {
```



```

//Test for a conflict with another real point.
if (belongs(0, t, truePoints))
{
    //replace the real point if the parametrization says yes.
    if (this.ruleTruePoint.replace() == true)
    {
        this.model.setElement(this.solutions.getPotential(0),
        constructPosition(0, t));
    }
    //make a deflagration if the parametrization says yes.
    if (this.ruleTruePoint.defla() == true)
    {
        this.doDeflagration(deflagration, 0, t, r);
    }
    //continue to the next state
    continue;
}

//Recall that this point was put.
addTruePoint(0, t);

//Add the potential of the real point to the model
this.model.setElement(this.solutions.getPotential(0),
constructPosition(0, t));

//Add the deflagration to the model
this.doDeflagration(deflagration, 0, t, r); }

}

//Write this model to a file
this.model.writeValuesModel(fileName, this.solutionsGen,
this.maximum);
//END OF ALGORITHM
}

```

8.4 Deflagration package

We recall that the aim of the Deflagration package is to draw the deflagration around a newly put point.

```

public class Deflagration
//calculates the deflagration for one given point.
{
    private int range;
    private int minimum;
        private int maximum;
        private int value;
        private int numberOfDim;
        private Vector point;
        private Vector position;
        private Vector sizeOfModel;
        private Model model;
    private Vector truePoints;
    private Vector initial;
    private FunctionHollow functionHollow;
    private RuleConflictDeflaHollow rule;

```

```

        public void setMinimum(int min)
        public void setMaximum(int max)
    public void setValue(int v)
        public void setNumberOfDim(int num)
        public void setPoint(Vector p)

```

```

        public void setSizeModel(Vector size)
        public void setModel(Model m)
        public void setRange(int r)
        public void setTruePoints(Vector tP)
    public void setFunctionHollow(FunctionHollow fctHollow)
    public void setRuleConflictDeflaHollow(RuleConflictDeflaHollow r)
    void

    /**
    ***ALGORITHM***
    */
    public void makeDeflagration()
    {
        initialPoint(); //set the point in the range to begin
        the deflagration

        while (nextPoint()) //while there's a next point in the
        range
        {
            putValueForThisPoint(); //put the
            right value in the model
        }

        return;
    }

```

```

        private void putValueForThisPoint()
        //For one suitable point in the range, put the right value in
        the model.
        {
            //calculate the value that would have to be put
            int posValue =
            this.functionHollow.function(this.range, this.value, this.point,
            this.position, this.sizeOfModel, this.minimum, this.maximum);

            //construct a position for the model
            Position pos = constructPositionOfPoint();

```

```

        //Conflict between deflagration and hollow to be resolved.
        if (this.model.getElement(pos).getValue() != -1)
        {
            posValue=this.rule.applyRule(posValue,
            this.model.getElement(pos).getValue());
        }

```

```

        //add the point to the model
        addPointToModel(posValue, pos);

        return;
    }

```

8.5 Model representation file format

The space generation process result is a text file, that can be read by decisional agent. We report here the basic structure of that file.

Dimensions:3;

Parameter:activity;
Dimension:3;

Parameter:insuline;
Dimension:2;

Parameter:glucose;
Dimension:1;

Elements in dimension 1:70;
Elements in dimension 2:60;
Elements in dimension 3:30;

Scale: 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73;
74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89;
90; 91; 92; 93; 94; 95; 96; 97; 98; 99; 100; 101; 102; 103;
104; 105; 106; 107; 108; 109; 110; 111; 112; 113; 114; 115;
116; 117; 118; 119; 120; 121; 122; 123; 124; 125; 126; 127;
128; 129; 130;
Scale: 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16;
17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32;
33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48;
49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60;
Scale: 0; 2; 4; 6; 8; 10; 12; 14; 16; 18; 20; 22; 24; 26; 28;
30; 32; 34; 36; 38; 40; 42; 44; 46; 48; 50; 52; 54; 56; 58; 60;

MODEL:
100, 100, 100, 100, 100, 100, 100, 100,
100, 100, 100, 100, 100, 100, 100, 100,
..., ..., ...

9 Appendix C: Model Visualization package

We have built a package to view the slice that are presented in Test section (Section 5). This package contains an applet, *Applet1*, that opens the file the user wants; later the Applet1 shows a 2D section of model.

```
public void init() {  
    try  
    {  
        System.out.println("Loading the model");  
        ModelBox mb = new ModelBox  
        ("C:/JBuilder3/myclasses/Model_4_GIdG.mod");  
  
        System.out.println("Model loaded.");  
  
        this.m = mb;  
        jbInit();  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```

References

- [1] A. Aarsten, D. Brugali, G. Menga, Designing Concurrent and Distributed Control System, Communication of the ACM, Oct., 1996.
- [2] F. Amigoni, N. Gatti, M. Somalvico, A Multiagent Interaction Paradigm for Physiological Process Control, 2001, submitted by AAMAS 2002.
- [3] F. Balduzzi, D. Brugali, A Hybrid Software Agent Model for Decentralized Control, Proceedings of the 2001 IEEE, International Conference on Robotics & Automation, Seoul, Korea – May 21-26, 2001.
- [4] M. Dini, N. Gatti, Agenzia Antropica: un sistema a molti agenti per applicazioni di natura biomedica, PM-AI&R Project, Politecnico di Milano, graduation thesis.
- [5] P. A. Fishwick, N. H. Narayanan, J. Sticklen, A. Bonarini, A multimodel approach to reasoning and simulation, Systems, Man and Cybernetics, IEEE Transactions on , Volume: 24 Issue: 10 , Oct. 1994, Page(s): 1433 -1449.
- [6] K. S. Fu, R. C. Gonzales, C. S. G. Lee, Robotics: control, sensing, vision, and intelligence. McGraw Hill, Book Company, 1987.
- [7] F. Heyes-Roth, L. D. Erman, A. Terry, B. Hayes-Roth, Domain-Specific Software Architectures: Distributed Intelligent Control and Management, IEEE 1991.
- [8] J. E. Larsson, B. Hayes-Roth, Guardian: An Intelligent Autonomous Agent for Medical Monitoring and Diagnosis, IEEE Intelligent Systems, Jan./Feb. 1998, pp 58-64, 1998.
- [9] W. Shen, D. H. Norrie, "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey", Knowledge and Information Systems: an International Journal, 1(2): 129-156, 1999.
- [10] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, Cambridge, MA, USA, 1999.